Aalto University
School of Science

Jani Suomalainen

Smart Authentication and Authorization in Heterogeneous Networked World

Thesis submitted in partial fulfillment of the requirements for the degree of Licentiate of Science in Technology.

Espoo, January 7th 2013

Supervisor: Prof. Tuomas Aura

Aalto University
School of Science

Aalto University School of Science Department of Computer Science and Engineering	ABSTRACT OF THE LICENTIATE THESIS

Author: Jani Suomalainen

Title: Smart Authentication and Authorization in Heterogeneous Networked World

Number of pages: 171	Date: January 7th 2013	Language: English
Professorship: Data Communications Software		Code: T008Z

Supervisor: Tuomas Aura

Instructor(s): Tuomas Aura

Abstract:

Our living environments are full of various connected computing devices. These environments in homes, offices, public spaces, transportation etc. are gaining abilities to acquire and apply knowledge about the environment and its users in order to improve users' experience in that environment. However, before smart adaptive solutions can be deployed in critical applications, authentication and authorization mechanisms are needed to provide protection against various security threats. These mechanisms must be able to interoperate and share information with different devices.

The thesis focuses to questions on how to facilitate the interoperability of authentication and authorization solutions and how to enable adaptability and smartness of these solutions. To address questions, this thesis explores existing authentication and authorizations solutions. Then the thesis builds new reusable, interoperable, and adaptive security solutions.

The smart space concept, based on semantic web technologies and publish-and-subscribe architecture, is recognized as a prominent approach for interoperability. We contribute by proposing solutions, which facilitate implementation of smart access control applications. An essential enabler for smart spaces is a secure platform for information sharing. This platform can be based on various security protocols and frameworks, providing diverse security levels. We survey security-levels and feasibility of some key establishment protocols and solutions for authentication and authorization. We also study ecosystem and adaptation issues as well as design and implement a fine-grained and context-based reusable security model, which enables development of self-configuring and adaptive authorization solutions.

Keywords: security, authentication, authorization, interoperability, smart space

Aalto University
School of Science

Aalto-yliopisto Perustieteiden korkeakoulu Tietotekniikan laitos	LISENSIAATIN TUTKIMUKSEN TIIVISTELMÄ

Tekijä: Jani Suomalainen

Nimi: Älykäs tunnistauminen ja käyttöoikeuksien hallinta monimuotoisessa verkotetussa maailmassa

Sivumäärä: 171	Päivämäärä: 7.1.2013	Kieli: Englanti
Professuuri: Tietoliikenneohjelmistot		Koodi: T008Z

Valvoja: Tuomas Aura

Ohjaaja(t): Tuomas Aura

Tiivistelmä:

Ympäristöt, joissa elämme, ovat täynnä erilaisia verkkolaitteita. Nämä koteihin, toimistoihin, julkisiin tiloihin ja ajoneuvoihin muodostuvat ympäristöt ovat oppimassa hyödyntämään ympäriltä saatavilla olevaa tietoa ja sopeuttamaan toimintaansa parantaakseen käyttäjän kokemusta näistä ympäristössä. Älykkäiden ja sopeutuvien tilojen käyttöönotto kriittisissä sovelluksissa vaatii kuitenkin tunnistautumis- ja käyttöoikeuksien hallintamenetelmiä tietoturvauhkien torjumiseksi. Näiden menetelmien pitää pystyä yhteistoimintaan ja mahdollistaa tiedonvaihto erilaisten laitteiden kanssa.

Tämä lisensiaatin tutkimus keskittyy kysymyksiin, kuinka helpottaa tunnistautumis- ja käyttöoikeusratkaisujen yhteensopivuutta ja kuinka mahdollistaa näiden ratkaisujen sopeutumiskyky ja älykäs toiminta. Tutkimuksessa tarkastellaan olemassa olevia menetelmiä. Tämän jälkeen kuvataan toteutuksia uusista tietoturvaratkaisuista, jotka ovat uudelleenkäytettäviä, eri laitteiden kanssa yhteensopivia ja eri vaatimuksiin mukautuvia.

Älytilat, jotka perustuvat semanttisten web teknologioiden ja julkaise-ja-tilaa arkkitehturin hyödyntämiseen, tunnistetaan työssä lupaavaksi yhteensopivuuden tuovaksi ratkaisuksi. Tutkimus esittää ratkaisuja, jotka helpottavat älykkäiden tunnistautumis- ja käyttöoikeuksien hallintaratkaisujen kehitystä. Oleellinen yhteensopivuuden mahdollistaja on tietoturvallinen yhteensopivuusalusta. Tämä alusta voi perustua erilaisiin avaintenhallinta ja tunnistautumisprotokolliin sekä käyttöoikeuksien hallintakehyksiin. Tutkimuksessa arvioidaan joidenkin olemassa olevien ratkaisujen käytettävyyttä ja tietoturvatasoa. Tutkimuksessa myös tutkitaan ekosysteemi- ja sopeutumiskysymyksiä sekä toteutetaan hienojakoinen ja kontekstiin perustuva uudelleen käytettävä tietoturvamalli, joka mahdollistaa itsesääntyvien ja mukatuvien käyttöoikeuksien hallinta sovellusten toteuttamisen.

Avainsanat: tietoturva, tunnistautuminen, käyttöoikeuksien hallinta, yhteensopivuus, älytila

Preface

I have been working in VTT Technical Research Centre of Finland since 2000. In my time in VTT, I have been working in various projects, which have been focusing on different security and access control related research questions. In my work I have been trying to focus on the management of the heterogeneity and complexity of information security solutions. This thesis is based on the work and articles done during years 2006 to 2012. In these years, I have been working on different projects, including HomeServices, ANSO, InterDeviceNoTA, Interceptor, and SOFIA, and focusing to home network and smart space related security technologies. The story for this thesis, the clue between the heterogenous articles, was drafted in Christmas holidays 2011 and the thesis was then finalized during the spring 2012.

I want to thank my colleagues in VTT who have been participating in to the same projects and who have been preparing the articles with me, including Antti Evesti, Pasi Hyttinen, Kari Keinänen, Juha Koivisto, Mika Rautila, and Eila Ovaska.

The thesis was made as a part of my studies in Aalto University. I want to thank Professor Tuomas Aura, who supervised the thesis. I also want to thank Professor Valtteri Niemi from University of Turku for preliminary examination of the thesis as well as Professors N. Asokan, Antti Ylä-Jääski and Kirsi Valtari for guidance during my post graduate studies.

Espoo, January 7th 2013

Jani Suomalainen

Table of contents

1 Introduction

1.1 Authentication and Authorization in Networked World

The amount of different networked devices and services has been rapidly increasing in the last decades. In the physical World, where we live in, we have seen networked sensors, cameras, video recorders, high definition televisions, PCs, printers, mobile phones, navigators, game consoles, and climate control equipment. In the virtual World, there is an enormous amount of information and different services available in remote servers. This development has also introduced various security threats. To protect us from these threats, we need different security technologies, including solutions for authentication and authorization.

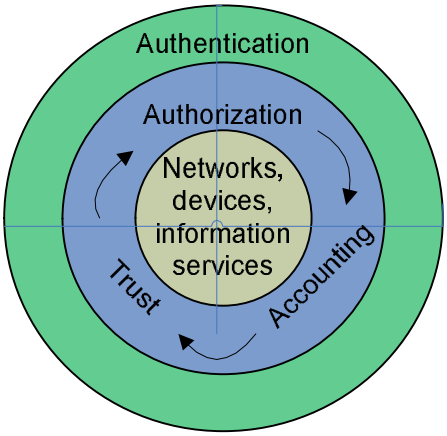

Figure 1. Core security enablers in the networked digital world

Authentication is a process of confirming an identity or an origin of a communication partner or a piece of information. Authentication makes it possible for an entity to verify that it really is interacting with those users and devices and downloading software from those servers it believes it is interacting with. Hence, authentication prevents misbehaving devices and users from providing bogus information. Authentication is a vital part or our everyday life and present, for instance, when making phone calls, when using a wireless headset, when watching a pay television, when opening electronic locks in an office, or when doing transactions within Internet banks. The cornerstone of authentication is the *establishment of cryptographic keys* between devices. Established keys can then be used with cryptographic security protocols to prove the authenticity of

information. Different key establishment and management mechanisms as well as security protocols can be applied in different environments starting from personal and home networks to ubiquitous systems and global Internet.

Authorization is a process of deciding whether an entity should be allowed to perform a particular action. In computer and communication systems, authorization mechanisms control and limit the risks caused by misbehaving users, devices or software components. A typical motivation for authorization is confidentiality, which is a principle ensuring that information is accessible only for authorized parties. Technically, authorization can mean a decision to establish a shared cryptographic key between devices. It can also mean a definition of detailed and complex security policies, specifying how different parties can cooperate in different situations. Authorization can be based on authentication in which case the authorization is given for known and *trusted* parties. Authorization is related to *accounting*, which refers to a process where users' actions are monitored. We make authorization decisions daily, for instance, when allowing a paired mobile phone to synchronize with PC's calendar, when allowing downloaded software to access network interfaces, when allowing an Internet bank to transfer our money, or when allowing family members to access photographs in a file sharing server.

Authentication and authorization mechanisms are based on established technologies, designed in the past decades [1, 2]. However, even during the last decade new and innovative solutions have been developed and emerged into markets, making the authentication and authorization more user-friendly, cost-efficient, or secure. Authentication based on new biometrics [3], such as kinetics [4], graphics-based passwords [5], password-less pairing mechanisms [6, 7], as well as authorizations based on contexts [8, 9, 10], trusted computing [11], or reputation [12, 13] are examples of recent activities within these areas.

1.2 Heterogeneity and Smartness of Information Security Solutions

The amount of solutions and variations for authentication and authorization is large. This amount of solutions is explained by the factors illustrated in Figure 2. Firstly, the amount of different application and environment specific network technologies is large. Technologies for sensors, web, homes, cars, public spaces, for instance, have their own characteristics and security requirements, causing that authentication and authorization solutions must be specifically crafted for them. Secondly, for one technology there are often various alternative security solutions. Typically one solution cannot provide the best usability, the best cost-efficiency and the best security level at the same time. Instead, alternatives providing different combination and compromises are provided for users and developers with different preferences and needs. Thirdly, different developers and manufacturers fulfil the requirements of applications and environments in their own manner and also provide unique and custom services. Gadgets services designed for special purposes and applications may provide unique solutions and utilize security solutions in their own ways. Standardisation efforts may ease the interoperability in some applications but the standards cannot cover all issues. Fourthly, new security solutions and improvements are constantly emerging as new ideas and security vulnerabilities are detected. New solutions are adopted but at the same time legacy systems continue their life alongside.

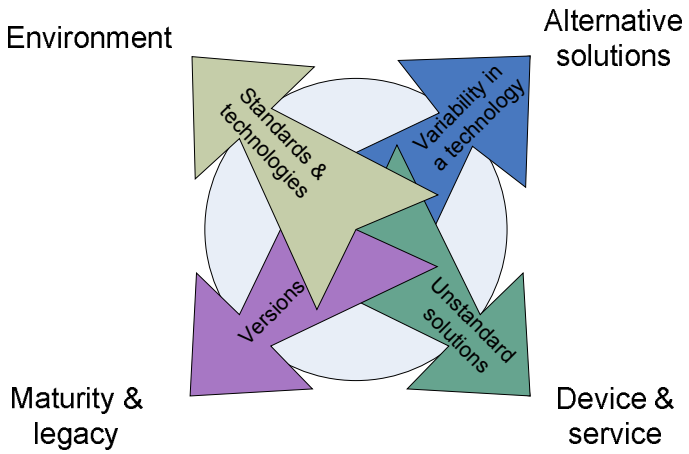

Figure 2. Causes of heterogeneity

This diversity and variability provides opportunities as well as challenges. Devices supporting incompatible authentication mechanisms cannot authenticate with each other. Authorization systems cannot be distributed and do not scale, when devices do not share common authorization solutions. In order to support each other devices must be equipped with several security mechanisms, which require additional hardware or which require software integration work and consume memory.

On the other hand, when the interoperability challenges can be solved, it is possible to select and adjust mechanisms so that the best possible mix between usability, cost efficiency and security is achieved in each particular authentication and authorization situation. Consequently, authentication and authorization mechanisms can be visualized as a part of concepts of smart environments, smart spaces, smart cities, and smart homes. In these concepts, the **smartness** *is defined as ability to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment* [14]. In this thesis, the **smart authentication and authorization** is *defined as an ability of an environment to acquire information and select mechanisms to provide authentication or authorization, which are the most suitable for a particular situation*. The smartness is based on solutions for autonomic computing [15] as well as on interoperability between authentication and authorization components. Essentially, the smartness comes from the intelligent management of authentication and authorization solutions and relevant information in an application or use case specific manner. Some examples of smart applications are presented in Section 6.

1.3 Interoperability Solutions for Authentication and Authorization

Authentication and authorization solutions can be based on diverse actors and components, forming an ecosystem. Particularly, an ecosystem consists of end-users, services, and third-party service providers; as well as of hardware and software components in the *secure interoperability platform*. The platform enables secure and authenticated communication between distributed devices belonging to different actors. The platform enables entities to understand each other and provides supportive services.

Figure 3 illustrates five strategies that the interoperability platform can utilize to achieve or enable interoperability. Depending on the actors and on the components in the ecosystem, different kinds of authentication and authorization applications can be set-up. Some solutions require direct realtime communication between communicating parties. In some solutions interoparabilyt and adaptation can be achived by delivering information only inderectionally or only in one direction.

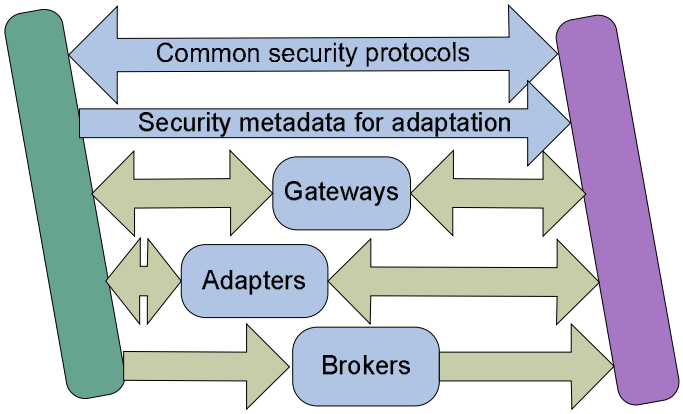

Figure 3. Strategies for enabling interoperability in information security systems

Networked entities can communicate and authenticate with each others when they share a common language i.e. when they use *common security protocols*. In the past, the standardization has been very successful in solving interconnectivity issues in communication protocols. Particularly, standardization is effective in the lower protocol levels, which are layers 1 to 6 (physical, data link, network, transport, session, and presentation) in the Open Systems Interconnection (OSI) model [16, 17]. Consequently, there exists large amount of standards for communication protocols suitable for various networks and devices. However, the standardization of the application-level interoperability, in OSI layer 7, is more challenging task [18].

In the application level, the amount of use cases and applications is huge and new applications are introduced rapidly. Standardisation takes time and, therefore, is not a sufficient answer to every possible need and use case. In the application level, one current trend has been the emergence of semantic web technologies [19, 20]. These standards are used to present application specific ‘languages’. A language is called an ontology and it presents concepts and relationships between concepts. Two parties do not need to know exactly the same language. Cooperation is successful when the

meanings of those concepts, which are essential for an application, are shared. Hence, introduction of new applications and application versions is easier, even though the approach does not guarantee interoperability.

An important part of interoperability is the devices' ability to adapt their behaviour to match to the requirements of their counterpartiers. The security handshake is a standard practise in security protocols. In handshakes, devices exchange security interface descriptions and other *security metadata.* This security metadata can contain information on the supported protocols, versions, or algorithms. Servers may deliver information on their privacy and security practises. The objective of negotiation may be to find the most secure or most optimal interoperable solutions. Further, the metadata can also contain instructions or policies on how the other party should behave and e.g. protect the delivered information. Both interacting devices may share metadata or only one may provide (in Figure 3 the fact that only another party needs to share is illustrated with one directional arrow).

Gateways and *adapters* are network elements, which enable communication between devices without shared communication protocol by automatically converting messages [21]. Adapters are device specific components enabling particular device to access network and gateways are generic components that can be used to enable communication between various devices. Adapters can be often considered as trusted components and can therefore authenticate peers on behalf of the adapted device. Gateways, which are used by many devices, may not be trusted to perform authentication actions on behalf of the devices. However, gateways can have a role when the converted information is not security critical.

Information *brokers* enable indirect collecting, sharing and delivery of information, which would not otherwise be available. Brokers can process and deliver information for authentication and authorization decisions. Broker can enable one or two directional unicast communication as well as multicasting. Brokers enable realtime communication and storing of information for later hadling. As brokers are central components they are able to control and authorize who can access brokered information and, thus, they are also able to enforce authorization policies. An example of a broker is a web reputation

service, which can be managed by a security company or community. These trusted third-party services collect and provide information on the trustworthiness of web services. Brokers have also a central role in the smart space concept. In smart spaces, communication between heterogeneous and ubiquitous devices is enabled by brokering information, structured according to semantic data presentation formats. Further, Certification authorities, the backbone of the internet security, can be also seen as brokers as they deliver vouched identity information to clients with certificates.

1.4 About this Thesis

1.4.1 Research Questions

The thesis concentrates on studying the heterogeneity of authentication and authorization solutions. The thesis explores both the opportunities as well as the challenges caused by the diversity and variability of devices, communication technologies, and applications. We will study the interoperability solutions in the protocol, platform and application levels. We will also focus on the enablers of smart authentication and authorization i.e. on solutions which enable finding, selecting and using authentication and authorization mechanisms and information in a manner which is the most suitable for each particular situation. Essential questions motivating the thesis include the following.

1. *How to facilitate interoperability of authentication and authorization solutions?*

Particularly, how to effectively utilize open standards to make interoperable security solutions? What is missing from the standards? This thesis concentrates on some standards, which are widely adopted and used typically by non-expert users. In the connectivity and network level, the surveyed standards include, e.g., Bluetooth, Wireless Fidelity (Wi-Fi), Transmission Layer Security (TLS) and Hypertext Transfer Protocol Secure (HTTPS). We will also cover presentation and middleware level technologies including Universal Plug and Play (UPnP), Device Interconnect Protocol (DIP) and Semantic Web. A particular focus is in the effectiveness of security mechanisms in these standards i.e. in the security level they provide.

The security levels that variable components can provide are different. In order to control this diversity and make sure that the overall security level reaches the minimal requirements, we need means to formally measure security levels provided by separate components. The thesis will study some metrics for quantifying the security levels of security systems. Later on the thesis also studies how to manage different security levels and enable systems to automatically select the mechanisms to provide the best suitable security level. The research question to be considered is:

2. *How do the solutions managing heterogeneity affect to actual security level and to users' perception of security and privacy?*

Particularly, we will focus on two questions. Firstly, how does the heterogeneity affect to strength and applicability of key establishment protocols? Secondly, how to estimate the impact that authentication mechanisms have on end-users' willingness to trust and authorize a communicating counterparty?

Heavy (and slow) standardization is not a viable solution in higly heterogenous and rapidly evolving application environments. To address the challenges and requirements caused by heterogeneity and complexity of authentication and authorization solutions in smart environments, the thesis constructs solutions for secure interoperability platform. The thesis presents case studies on how to use adapters, brokers, or semantic technologies to achieve smart and interoperable security solutions. The essential research question is:

3. *How to build facilities for smart access control applications using a combination of brokers and middleware approaches?*

In more detail the thesis studies, how do these solutions increase interoperability and security? What kind of building blocks and ecosystems are needed? How to increase the reusability, flexibility, and security level of these solutions? What kind of challenges does the deployment of these solutions cause?

1.4.2 Research Methods

The research method applied in this thesis can be characterised as literature-based constructive research. The thesis builds on literature survey where authentication and

authorization solutions are presented, studied, classified, and analysed. The security and feasibility analyses over existing mechanisms are mainly qualitative. However, quantitative research methods are also used to examine the impact of authentication solutions for end-users' perception of security and privacy. Particularly, the qualitative analysis is used to study a correlation between databases containing SSL certificates, popularity information of web services, and web reputation information of web services. This analysis provides input for constructing solutions and mechanisms.

The constructive research [22] tests theories and proposes novel solutions to practically and theoretically relevant problems. The approach is to build artefacts - such as models, methods, and algorithms - to create knowledge on how to solve the problem. In constructive research, first an understanding of the problem is gained, then artefacts are constructed and demonstrated, and finally the theoretical connections and applicability are examined. In this thesis, the focus is in on two new artefacts: OpenHouse and RDF Information Base Solution (RIBS). The thesis will describe security approaches for these middleware and service platform solutions.

1.4.3 Publications

The results presented in this thesis have been previously published and validated in the peer reviewed conference and journal articles. The articles have been restructured and rewritten to form a backbone for this thesis. The articles are the following:

I Jani Suomalainen, Jukka Valkonen, N. Asokan. *Standards for Security Associations in Personal Networks: A Comparative Analysis*. International Journal of Security and Networks (IJSN). Vol. 4, Nos. 1/2. Pp. 87–100, February 2009. Inderscience[1]. (A preliminary version published in Proceedings of the European Workshop on Security and Privacy in Ad-hoc and Sensor Networks (ESAS 2007) [23][2].)

[1] Copyright Inderscience. Excerpts and illustrations reprinted with kind permissions.
[2] Copyrigt Springer Science and Business Media. Excerpts and illustrations reprinted with kind permissions.

II Jani Suomalainen. *Towards Fine-Grained Authorizations in Small Office and Home Networks*. Proceedings of the Second International Conference on Systems and Networks Communications (ICSNC 2007), Cap Esterel, French Riviera, France. 25-31 August 2007. IEEE Computer Society[3].

III Jani Suomalainen, Seamus Moloney, Juha Koivisto, Kari Keinänen. *OpenHouse: a Secure Platform for Distributed Home Services*. Proceedings of the Sixth Annual Conference on Privacy, Security and Trust (PST 2008). Fredericton, New Brunswick, Canada. 1-3 October 2008. Pp. 15-23. IEEE Computer Society[3].

IV Jani Suomalainen, Pasi Hyttinen, Pentti Tarvainen. *Secure Information Sharing between Heterogeneous Embedded Devices*. The First International Workshop on Measurability of Security in Software Architectures (MeSSa 2010). Proceedings of the Fourth European Conference on Software Architecture: Companion Volume. Copenhagen, Denmark. 23 August 2010. Pp. 205-212. ACM[4].

V Jani Suomalainen. *Flexible Security Deployment in Smart Spaces*. The First International Workshop on Self-managing Solutions for Smart Environments (S3E 2011). Oulu, Finland, 11 May 2011. Proceedings of the 6th International Conference on Grid and Pervasive Computing (GPC2011) Workshops. Lecture Notes in Computer Science, Vol. 7096. Springer[2].

VI Jani Suomalainen and Pasi Hyttinen. *Security Solutions for Smart Spaces*. The Second International Workshop on Semantic Interoperability for Smart Spaces (SISS2011). Proceedings of 2011 IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2011). Munich, Germany. 18-22 July 2011. Pp. 297-302. IEEE Computer Society[3].

VII Jani Suomalainen. *Quantifying Value of SSL Certification with Web Reputation Metrics*. Proceedings of the Seventh International Conference on Internet

---

[3] Copyright IEEE Computer Society. Excerpts and illustrations reprinted with kind permissions.

[4] Copyright ACM. Excerpts and illustrations reprinted with permissions. http://doi.acm.org/10.1145/1842752.1842793.

Monitoring and Protection (ICIMP 2012). Stuttgart, Germany. May 27 - June 1, 2012. Pp. 7-12. XPS[5].

Articles I, II, and III study how authentication and authorization technologies can be applied in personal and home networks. Article I describes taxonomy of protocols for key establishment between personal devices and analyses use of key establishment mechanisms in emerging standards. The author contributed in the survey of standards, participated in the security analysis of key establishment mechanisms, and described novel man-in-the-middle attacks. Article II describes authorization requirements for home network middleware and proposes a conceptual model ('permission attenuation') for managing authorizations in systems with cooperative components. In Article III, a secure service platform and authorization model, which considers whole ecosystem for home services and makes security configuration in homes more usable, is presented. The author was the main designer and sole implementer of the authentication and authorization characteristics for the proposed OpenHouse platform.

Articles IV, V, and VI address the security issues in smart spaces. Smart space is brokered information sharing environment, which facilitate self-adaptability and interoperability between ubiquitous devices with semantic web technologies. In Article IV, security architecture and a mechanism for controlled information sharing between devices with heterogeneous security properties is described. Article V addresses security configuration of smart spaces and presents an example of self-configuring using role-based access control. The article also shows how smart space brokers can be used as mediators of key establishment between heterogeneous devices. Article VI describes our security implementations for smart spaces. The article presents security requirements in different conceptual layers of semantic web and smart spaces. The article provides introduction to the proposed an access control framework for the implemented RDF information base solution. The article also gives examples on building of self-adaptive and context based security solutions by using rule based reasoning. The author of this thesis was the main designer and implementer of security functionality for the broker implementation and for supporting communicating libraries.

---

[5] Copyright (c) IARIA, 2012. Excerpts and illustrations reprinted with permissions.

Article VII provides an example on the use of more rich security information in constructing smart authorization solutions. The article concentrates to the authentication and authorization in the Internet and World Wide Web. Article VII, which was written completely by the author, explores fine-grained reputation information of the security characteristics of web services. The paper studies the potential of web reputation as a universal security metric for web servers and provides statistical analysis on the correlation between reputation and SSL certification.

1.4.4 Organisation

The thesis is organized as follows. Section 2 studies the heterogeneity of authentication capabilities within different connectivity mechanisms and physical interfaces, available for personal devices. Particularly, the section surveys recent standards and key establishment mechanisms proposed for personal devices. The section notes how different usability, security level, and cost requirements cause variability and proposes novel mediator based protocols for easing interoperability.

Section 3 surveys authentication and authorization solutions for large environments. The section focuses on internet security solutions, where additional security data, structured according to few common standards, is provided to clients, which are connecting to servers. The solutions are SSL certification with extended validation and web reputation. Novel contribution in the section is the proposal of reputation correlation metric for analysing impacts of security mechanisms. The section studies the correlation between SSL certifications and reputations of web servers.

Section 4 surveys authentication and authorization requirements from the point of view of home networks. The section studies what security mechanisms are needed and available in existing network frameworks. Further, the section contributes by describing experiences with a secure middleware platform implementation, called OpenHouse.

Section 5 studies how semantic interoperability solutions can be applied to provide an interoperable and secure platform for ubiquitous networks. The section will represent the concept of smart spaces and survey security requirements within semantic web technologies. Then, the section contributes by presenting design and implementations of authentication and authorization mechanisms for an interoperability platform. The

platform is based on our semantic information broker implementation, called the RDF Information Base Solution (RIBS), supporting a reusable, fine-grained and context-aware access control model.

Section 6 presents application examples of smart authentication and authorization. The section presents smart applications making authorizations in home and ubiquitous environments autonomous and self-adaptable. The applications are based on the platform presented in Section 5.

Section 7 discusses on the significance of the results. Particularly, the section answers to the reseach questions, which were given in this Introduction section, and presents unanswered research questions and areas for the future research.

Section 8 concludes the paper by listing the key contributions made in this thesis.

2 Heterogeneity in Key Establishment Protocols

Introducing a new device to a network or to another device is one of the most security critical phases of communication in personal networks. It is difficult to make this process of associating devices easy-to-use, secure and inexpensive at the same time. A cornerstone of this process is key establishment. There have been a number of research proposals for key establishment in personal networks. Some of them have been adapted by emerging standard specifications. In this section, we first present taxonomy of protocols for key establishment in personal networks. Further, we describe and analyze specific protocols. We then use this taxonomy in surveying and comparing association models proposed in several emerging standards from security, usability, and implementation perspectives.

Subsection 2.1 studies the heterogeneity in key establishment protocols, which are used by personal devices. In Subsection 2.1, we will survey and analyze the existing mechanisms and standards, namely Bluetooth, Wi-Fi, Wireless USB and HomePlugAV. The survey is based on Article I.

Subsection 2.2 complements the analysis by focusing on interoperability challenge, which is caused by these emerging mechanisms, and by presenting a mediator concept and protocols for easing interoperability. Mediator devices are advanced devices such as mobile phones or PCs, which support several association interfaces. Particularly, the subsection proposes a mechanism for extending the Bluetooth Secure Simple Pairing standard to support associations through mediators. Essentially, the subsection describes how a Bluetooth device supporting any out-of-band association model can be paired with a device supporting incompatible out-of-band model, comparison model or passkey entry model (described in the following Subsection 2.1).

2.1 Key Establishment Mechanisms for Personal Devices

2.1.1 Introduction

Short-range communication standards have brought a large number of new services to the reach of ordinary users. For instance, standards for personal networking

technologies such as Bluetooth, Wi-Fi, Wireless Universal Serial Bus (WUSB) and HomePlugAV enable users to easily introduce, access, and control services and devices both in home and mobile environments.

The initial process of introducing a new device securely to another device or to a network is called, in this section, an association. Association consists of the participating devices finding each other and establishing a shared secret key between them.

The part of the association procedure that is visible to the user is called an association model Association models in today's personal networks such as those based on Wi-Fi or Bluetooth, typically consist of the user scanning the neighborhood from one device, selecting the other device or network to associate with, and then typing in a shared passkey. These current association procedures have several usability and security drawbacks arising primarily from the fact that they are used by ordinary non-expert users. First, when there are many devices or networks in the scanned neighborhood, users find it difficult to choose the correct one from a, possibly long, list of choices. Second, the security of the association protocol depends on the strength of the shared passkey. If passkeys are long and hard-to-guess, usability is impaired. Using a short or memorable passkey leaves the protocol vulnerable to dictionary attacks, even by passive eavesdroppers. Also, over the last few years several other weaknesses have also been discovered in the association protocols used in Wi-Fi and Bluetooth [24, 25].

To address these concerns, various new ideas have been proposed with the intent of providing a secure yet usable association model. For instance, there have been proposals for association models utilizing short passwords/checksums [26, 27, 28, 29, 30, 31] or various types of out-of-band channels [32, 33, 34, 35, 36]. However, in reality, it is impractical to mandate a single model for all kinds of devices because different devices have different hardware capabilities. Also, different users and application contexts have different usability and security requirements. Because of this, forthcoming standards are adopting multiple association models. Although, low-end devices like headsets and wireless access points may be limited to one association model, richer devices like mobile phones and personal computers will naturally support several. The security of

individual association models has been studied widely. But new kinds of threats may emerge when several models are supported in personal devices and several protocols and versions of protocols are in use simultaneously.

In this section, various protocols for key establishment and taxonomy for classifying them are presented. Then, association models proposed in different standards are comparatively analyzed from a practical point of view. The surveyed standards are Bluetooth Secure Simple Pairing [37], Wi-Fi Protected Setup [38], Wireless USB Association Models [39], and HomePlugAV security modes [40, 41]. The section reveals the similarities between the protocols in different standard specifications by relating them to the taxonomy. All of the surveyed standards are targeted for personal devices and support multiple association models.

The rest of this section is organized as follows. Subsection 2.1.2 provides a systematic taxonomy of different protocols for key establishment and describes some basic protocols. Subsection 2.1.3 look at how different types of secure channels and physical interfaces can be used to implement the protocols. Subsection 2.1.4 explains how and which key establishment protocols and related association models are used in the surveyed standards. Subsection 2.1.5 evaluates and analyzes the security of various key establishment models described in the standards. Then, new attacks against the methods, published in Article I, are described in Subsection 2.1.6.

2.1.2 Key Establishment Protocols

2.1.2.1 Classification of Key Establishment Methods

All of the association models we will survey in the following Subsection are based on one or more protocols for human-mediated establishment of a shared key between two devices. The shared key is typically used to protect subsequent communication over the otherwise insecure communication channel and, possibly, in authentication for other access control decisions. We show that the same basic protocols are used in different standard specifications, even though the exact instantiations naturally differ.

The attacker model for key establishment is the following. The two devices involved in key establishment are capable of communicating over an insecure communication

channel. The devices themselves are assumed to be secure and trustworthy. The attacker has the standard Dolev-Yao capabilities [42] over the insecure channel: the attacker can insert, delete, modify or delay messages sent over the insecure channel. The security objective of the participating devices is to establish a common key, which is shared only between the associated devices and which is used to protect subsequent communication between the devices. The goal of the attacker is to intervene in this process so that either it can read subsequent communication between the participating devices, or act as an active man-in-the-middle. In the latter case, the attacker can generate or modify messages and fool one or both of the devices into accepting these messages as originating from the peer device.

Figure 4 presents taxonomy of key establishment protocols that can be used to associate personal devices. At a high level, key establishment may be a simple *key transport* or involve running a *key agreement* protocol. In the context of personal networks where the devices are likely to be in close proximity, an additional key establishment method is *key extraction* from the common shared environment.

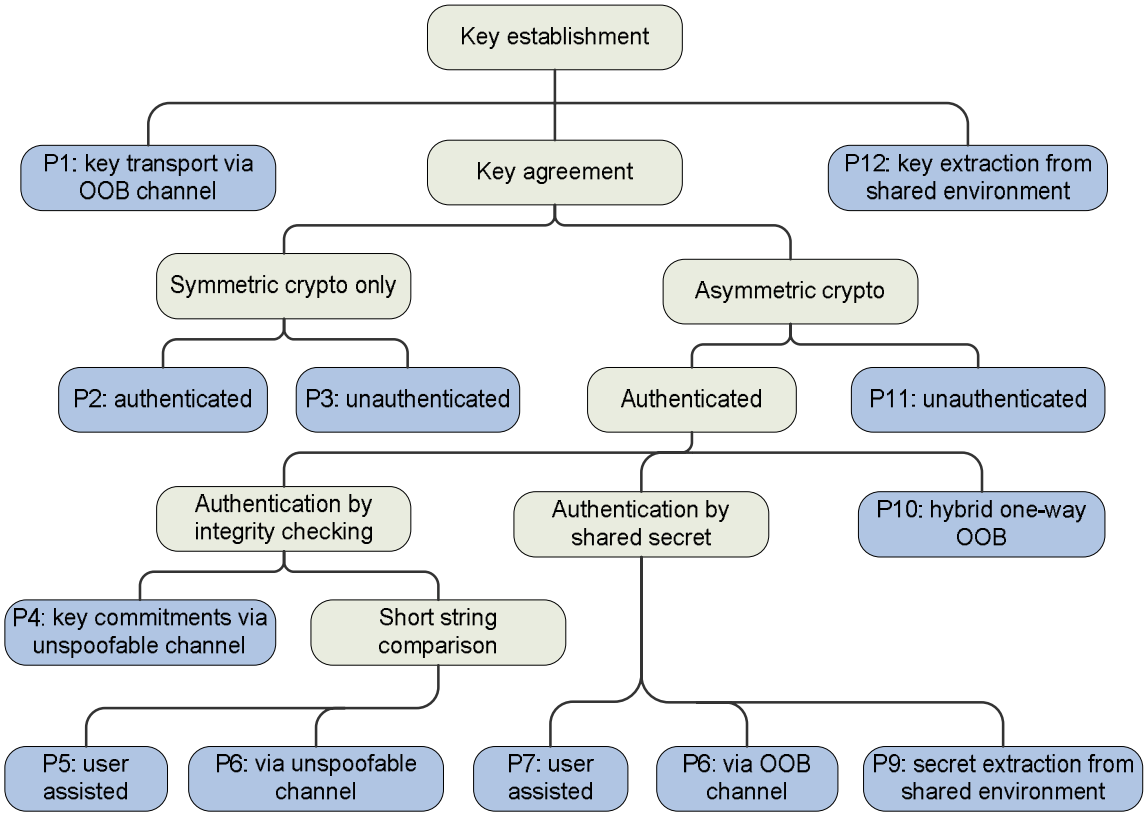

Figure 4. Taxonomy of key establishment methods [Article I]

Key transport: In key transport, one device chooses the key and transmits it directly to the second device using an out-of-band (OOB) secure communication channel (**P1**). Typical out-of-band channels used for key transport include a direct USB cable connection or the use of removable memory, like flash drives. The security of key transport depends on the out-of-band channel being secret and unspoofable: a man-in-the-middle must not be able to modify the data transmitted out of band between the devices.

Key extraction: Personal devices are often in close proximity to one another and thus share a common ambient environment. This gives rise to an interesting possibility for key establishment: measurements of certain environmental parameters, such as the signal strengths of radio beacons in the vicinity [43] or ambient noise, may be similar in devices that are close to each other but hard to predict from devices that are not in the same place at the same time. By measuring such parameters, and using them in a key agreement protocol, the devices may be able to *extract an authenticated shared secret* (**P12**).

Key Agreement: Key agreement protocols may be based purely on symmetric key cryptography, or may be based on asymmetric key cryptography as well. In the latter case, the typical protocol is the key exchange presented by [44]. Key agreement may be *unauthenticated* or *authenticated*. Unauthenticated symmetric key agreement (**P3**) is vulnerable even to passive eavesdroppers. Unauthenticated asymmetric key agreement (**P11**) is secure against passive eavesdroppers but is vulnerable to active man-in-the-middle.

2.1.2.2 Authentication methods

There are a number of ways to authenticate key agreement. Key agreement based on symmetric key cryptography is authenticated by using a sufficiently long *pre-shared secret* (**P2**). The security of such protocols depends on the length of the pre-shared secret. Authentication of asymmetric key agreement can be performed using some form of *integrity checking*, or by using a pre-shared secret or using a combination of these two. Authentication by integrity-checking can be done either by exchanging and comparing commitments to public keys, or by exchanging and comparing short integrity checksums.

Authentication by exchanging key commitments: A simple protocol to authenticate the public keys of two devices is to use an auxiliary channel to exchange commitments to the public keys (**P4**) [33]. The auxiliary channel is unspoofable in that it is difficult for an attacker to insert, modify or delete messages in the channel without being detected. When the devices exchange public keys via the in-band channel, they can validate the authenticity of these keys by using the information exchanged via the auxiliary channel.

The security of the protocols depends on the auxiliary channel being unspoofable and on the commitments of public keys being strong enough. There are two ways to realize such auxiliary channel. The first is to use a separate, out-of-band, physical channel which is resistant to spoofing. Several such out-of-band channels have been proposed in the literature including audio [45], visual [34, 35], infrared [33] and Near-Field Communication (NFC). Both devices involved in the association are assumed to support the same type of physical hardware interfaces. The second way is to use the *I-*

Codes [46] technique which uses the anti-blocking property inherent in some otherwise insecure in-band channels (In such channels the standard Dolev-Yao attacker model is too strong) to construct a logical auxiliary channel which is difficult to spoof. Commitments to public keys should be strong enough (e.g., a cryptographic hash function with at least 80 bits of output) to resist the attacker finding a second pre-image to the commitment.

Authentication by short integrity checksum: The idea of using short checksums to authenticate a key agreement was originally proposed in PGPfone [26]. Afterwards several researchers have proposed variations and enhancements [29, 30, 31, 47]. In these protocols, each device computes a short checksum from the messages exchanged during the key agreement protocol. As we shall see in the example protocol below, the messages are structured such that if the two checksums are the same, the exchange is authenticated. This is sometimes referred to as ``short authenticated string" (SAS) protocols. A basic three round mutual authentication protocol [29] is illustrated, in a simplified form, in Figure 5.

The notations are as follows: in practice, h is a cryptographic hash function like SHA-256; f is also a hash function, but with a short output mapped to a human-readable string of digits. The hat '^' symbol is used to denote the receiver's view of a value sent in protocol message over the insecure in-band channel.

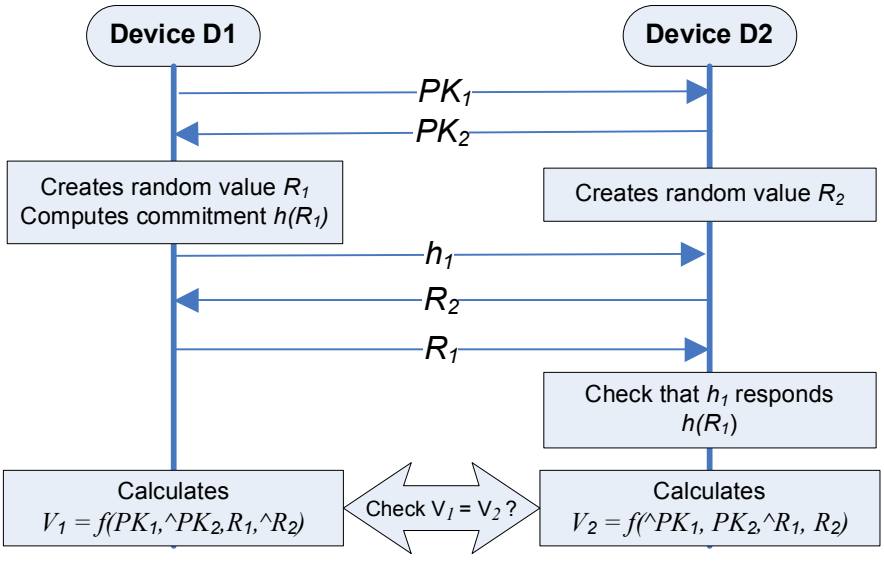

Figure 5. Authentication by short integrity checksum

The protocol steps are the following:

1. Devices D1 and D2 first exchange their public keys PK_1 and PK_2.
2. D1 generates a long random value R_1, computes commitment $h_1 = h(R_1)$ and sends it to D2
3. D2 generates a long random value R_2 and sends it to D1
4. D1 sends R_1 to D2
5. D2 checks if $\hat{}h$ equals $h(\hat{}R_1)$. If equality holds, D2 computes $V_2 = f(\hat{}PK_1\}, PK_2, \hat{}R_1, R_2)$, otherwise it aborts.
6. D1 computes $V_1 = f(PK_1, \hat{}PK_2, R_1, \hat{}R_2)$.
7. User checks if V_1 equals V_2

The check in the last step can be done in many different ways. One way is to ask the user to do the comparison (**P5**): Each device 'shows' its own string to the user and ask whether it is the same as what the other device is showing. 'Showing' can use any applicable user interface: displaying the string on a screen, or having a voice synthesizer read out the characters in the string. If the checksum strings are identical, the user indicates this to both devices and both devices conclude that the authentication is successful. Otherwise, the user indicates a mismatch to both devices and both conclude that the authentication did not succeed. An alternative way is to do the check using an auxiliary unspoofable channel (**P6**). The unspoofable channel can be a physical out-of-band channel, as presented by [35, 36], or an I-Codes channel by [46].

To break this protocol, a man-in-the-middle has to choose random numbers R'_1, R'_2 and public keys PK'_1, PK'_2 so that $f(PK'_1, PK_2, R'_1, R_2)$ equals $f(PK_1, PK'_2, R_1, R'_2)$. The security of the protocol depends on the quality of the functions h and f. If h is collision-resistant, the attacker has to choose R'_1 without knowing anything about R'_2. If h is one-way, attacker has to choose R'_2 without knowing about R'_1. If the output of f is a uniformly distributed n-bit value, then the chance of a man-in-the-middle succeeding is $1/2^n$. This success probability does not depend on any additional assumptions about the computational capabilities of the attacker beyond that he cannot break h in real time. The formal proofs were presented by [48].

Authentication by (short) shared secret: Key exchange can also be authenticated using a short pre-shared secret passkey. A number of different methods have been proposed for password-authenticated key exchange since the idea was introduced by [49]. In Figure 6 we describe a variant of the MANA III protocol by [28] originally described by [27]. It uses a one-time passkey P to authenticate PK_1 and PK_2. P is split into k pieces, labelled P_1 … P_k. The steps in the protocol are repeated k times. The figure shows the exchanges in the i^{th} round.

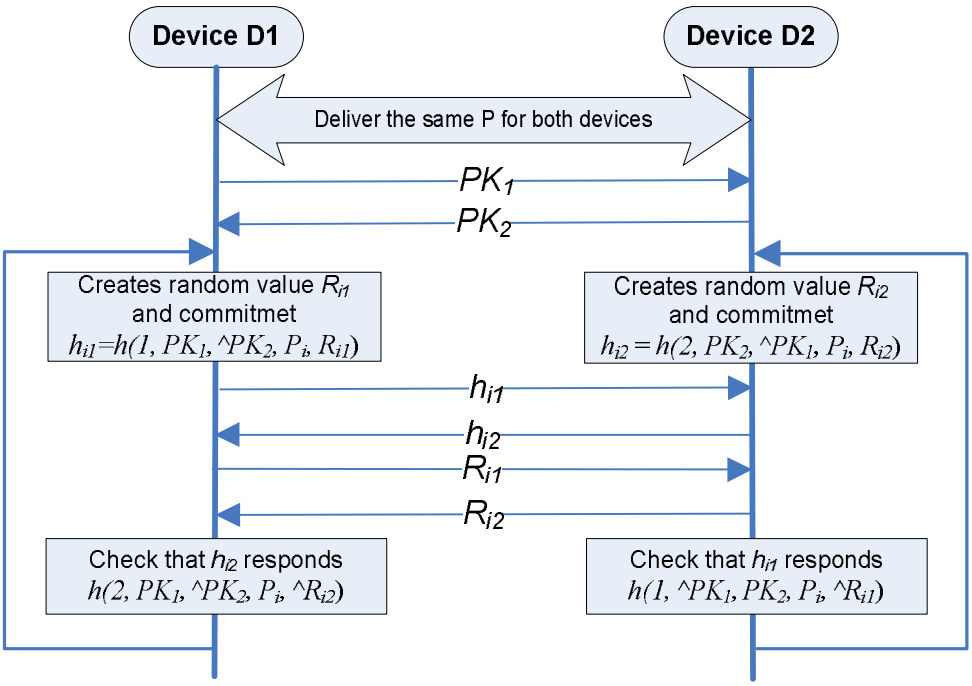

Figure 6. Round i of authentication by (short) shared secret

The protocol steps in each round are the following:

1. D1 generates a long random value R_{i1}, computes commitment $h_{i1} = h(1, PK_1, ^PK_2, P_i, R_{i1})$ and sends it to D2
2. D2 generates a long random value R_{i2}, computes commitment $h_{i2} = h(2, PK_2, ^PK_1, P_i, R_{i2})$ and sends it to D1
3. D1 sends a long random value R_{i1} to D2
4. D2 sends a long random value R_{i2} to D1
5. D2 checks if $^h_{i1}$ equals $h(1, ^PK_1, PK_2, P_i, ^R_{i1})$. If it does not hold, it aborts.

6. D1 checks if $\hat{h}_{i2}$ equals $h(2, PK_1, \hat{PK}_2, P_i, \hat{R}_{i2})$. If it does not hold, it aborts.

In each round, each party demonstrates its knowledge of P_i. A man-in-the-middle can learn P_i by sending garbage in message 2, and figuring out P_i by exhaustive search once D1 reveals R_{i1} in message 3. However, without knowing P_i, $i = 2 \ldots k$, the attacker cannot successfully complete the protocol run (recall that P is a *one-time* passkey). With n-bit passkey and k rounds the probability for a successful man-in-the-middle attack is $2^{-(n-(n/k))}$. As in the case of short authentication string, the man-in-the-middle success probabilities do not depend on additional assumptions about the attacker's computational capabilities.

There are three different ways for arranging for both devices to know the same P. One way is to have the user as the intermediary (**P7**): one device may show a value for P which the user is asked to enter into the second device, or the user may choose P and enter it into both devices. Alternatively, P may be transported from one device to another using an out-of-band channel providing communication secrecy (**P8**). A third possibility is to extract P from the shared environment (**P9**) [43]. In the latter two methods, there is no need for a human to transfer P between the devices. Consequently P can be longer, thus making probability for a successful attack smaller. Note that P is still used only to authenticate the key agreement, rather than as the long term secret.

Hybrid authentication: Hybrid authentication protocols are used to achieve mutual authentication when only a one-way out-band-channel is available (**P10**). The one-way channel is used to transmit the shared secret value and a hash of the public key from the first device to the second. The second device authenticates the first based on the public key hash. The first device authenticates the second based on its knowledge of the shared secret. A basic protocol is depicted in Figure 7.

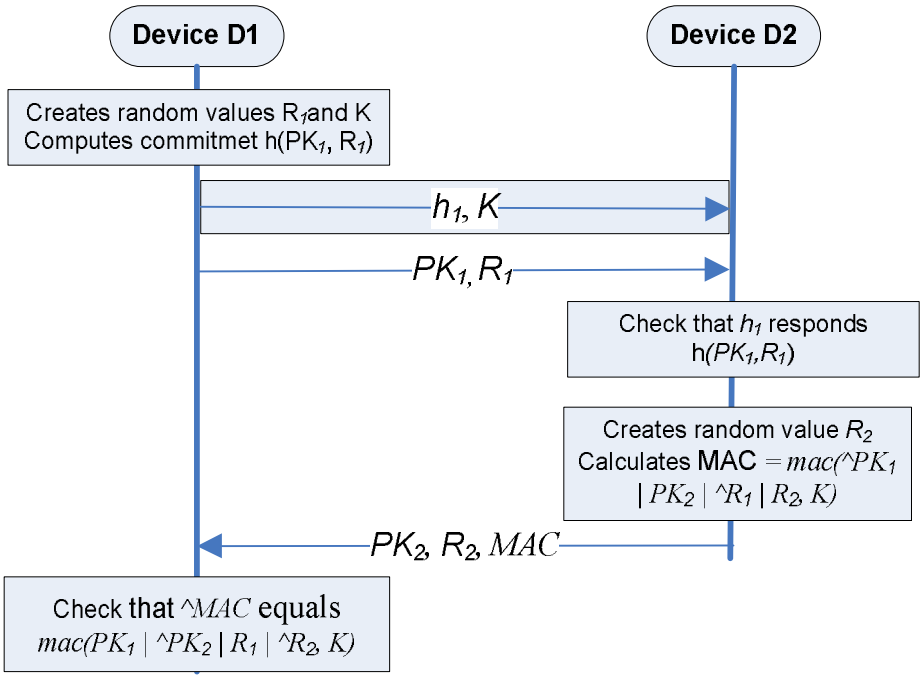

Figure 7. Hybrid authentication protocol

The protocol has the following steps:

1. D1 picks two long random values R_1 and K, computes commitment h to public key PK_1 as $h = h(PK_1, R_1)$ and sends h and K using OOB channel
2. D1 sends its public key and random value using in-band channel.
3. D2 checks if h equals h(^PK$_1$,^R$_1$) and aborts if it does not hold. Otherwise, D2 picks its own long random value R_2, computes message authentication code (MAC) using a key K. $MAC = mac(^PK_1 \mid PK_2 \mid ^R_1 \mid R_2, K)$ and sends the result to D2 with its own public key and random value.
4. D1 checks if ^MAC equals $mac(PK_1 \mid ^PK_2 \mid R_1 \mid ^R_2, K)$. If it does not hold, it aborts.

The security of the protocol depends on the out-of-band communication being both secret and integrity-protected, as well as on strength of the hash function h and the message authentication code function c.

2.1.3 Secure Channels and Physical Interfaces

In this section, we survey secure out-of-band communication channels and physical interfaces and how these channels can be used for key establishment in the various

methods we looked at in previous subsection. Out-of-band channels are communication channels distinct from the insecure channel over which the devices normally communicate. Using out-of-band channels to aid in association and key establishment can greatly improve usability by minimizing user actions. Therefore, researchers have looked for ways of using out-of-band channels in key establishment [32]. Various types of out-of-band channels have been considered in the literature including physical contact [32], infrared [33], audio channels [36], visual channels [34, 35] and, very short-range wireless communication channels like Near Field Communications (NFC). Different types of channels have different characteristics which affect their applicability to the different methods. The characteristics that are relevant for key agreement are the following:

1. **Channel security**: All useful types out-of-band channels are assumed to provide *integrity*: an attacker is assumed incapable of modifying, inserting or deleting messages sent via the channel. Some types are assumed to provide *secrecy* as well: an attacker is assumed incapable of reading the information sent via the channel. Usually physical connections and NFC channels are assumed to provide secrecy; however the validity of these assumptions have been questioned [50].

2. **Directionality**: Depending on the hardware available on the devices, the out-of-band channel may be unidirectional or bidirectional.

3. **Bandwidth**: Bandwidth of a channel is the rate at which it can transfer data. The bandwidth of an out-of-band channel is relevant in key establishment because it influences the time it takes to complete the association process.

Table 1 lists the protocols from Section 2.1.2 that can be implemented using out-of-band channels. The table gives also characteristics that these protocols require from out-of-band channels.

Table 1. Requirements that key establishment methods cause for of out-of-band channels [Article I]

Method	Integrity	Secrecy	Directionality	Data size
P1: Key transport		√	1-way	128-256

				bits
P4: Exchange of key commitments	√		2-way	128-256 bits
P6: Short string comparison	√		1-way	12-20 bits
P8: Transfer of (short) secret		√	1-way	12-20 bits
P10: Transfer of commitment and secret	√	√	1-way	128-256 bits

Although the promise of better usability is the motivation for using out-of-band channels in key establishment, the downside is the need to have the necessary hardware interfaces on both devices. There is no universal out-of-band channel guaranteed to be available on all devices. The vast majority of personal devices are low-cost commodity devices. Therefore adding a new hardware interface simply for the purpose of easing the association process is usually not an economically viable option. Researchers have therefore investigated ways to establish associations while maximizing security, usability and cost. One approach is to design the association procedures taking the resource asymmetry between the devices involved in the association. Typically one device, like a laptop or phone, has greater capabilities, while the other, like an access point or headset, is extremely resource constrained and cost-sensitive. Setting up a security association using a visual channel is described in [35]: one device is assumed to have a video camera while the other device needs to have only a single light source (such as a light-emitting diode) and mechanisms for user confirmation (like buttons for indicating yes and no).

Characteristics of in-band communication channels have been utilized by some key establishment protocols to strengthen security level. These schemes are based on the fact that signal quality is different in different locations. For instance, [40] observed that signals on power-line channel must be adapted for each receiver and because of that eavesdropper cannot receive good enough signal. Further, they argue that active online attacks can be easily detected in a narrowband power-line channel. Generation of shared keys from signal envelopes in wireless networks is proposed in [51].

2.1.4 Key Establishment Models in Standards

This section surveys the secure association models adopted to standards for communication with personal devices. The standards are compared by referring to the classification presented in Subsection 2.1.2.1.

2.1.4.1 Bluetooth Secure Simple Pairing

Bluetooth Secure Simple Pairing (SSP) [37] is intended to provide better usability and security than the original Bluetooth pairing mechanism, and is expected to replace it. Simple pairing consists of three phases. In the first phase, the devices find each other and exchange information about their user input/output capabilities and their elliptic curve Diffie-Hellman public keys. In the second phase, the public keys are authenticated and the Diffie-Hellman key is calculated. The exact authentication protocol, and hence the association model, is determined based on the device user-I/O capabilities. SSP supports four different association models: Numeric Comparison, Passkey entry, `Just Works' and Out-of-band models:

Numeric comparison model is for end-user's manual comparison and confirmation whether short integrity checksums displayed by both devices are identical (**Figure 4**: **P5**). The compared checksum is 6 digits long. The phase 2 protocol is an instantiation of the protocol in Figure 5. **Passkey entry model** is targeted primarily for the case where only one device has a display but the other device has a keypad. The first device displays the 6-digit secret passkey, and the end-user is required to type it into the second device. The passkey is used to authenticate the Diffie-Hellman key agreement (**Figure 4**: **P7**). The protocol is based on user-assisted authentication by shared secret in Figure 6 with 20 rounds (k=20). Devices prove knowledge of one bit of the passkey in each round.

1. **'Just works' model** is targeted for cases where at least one of the devices has neither a display nor a keypad. Therefore, unauthenticated Diffie-Hellman key agreement is used (**Figure 4**: **P11**) to protect against passive eavesdroppers but not against active man-in-the middle attacks.

2. **Out-of-band model** is intended to be used with different out-of-band channels, in particular with Near Field Communication technology. Device D_A uses the

out-of-band channel to send a 128-bit secret r_A and a commitment C_A to its public key PK_A. Similarly, D_B uses the out-of-band channel to send r_B and C_B. If the OOB channel is bidirectional, mutual authentication is achieved by each party verifying that the peer's public key matches the commitment received via the out-of-band channel. (**Figure 4**: **P4**) If the OOB channel is only one directional, the party receiving the out-of-band message can authenticate the public key of its peer. However, the party sending the out-of-band message must wait until the third phase of SSP to send a proof-of-knowledge of the shared secret *r*. (**Figure 4**: **P10**)

In the third phase of simple pairing, the agreed key is confirmed by exchanging message authentication codes using the newly computed Diffie-Hellman key. Each device includes the random value *r* received from the peer in the calculation of its message authentication code.

Peer discovery: In original Bluetooth pairing, peer discovery is left to the user: the user initiates pairing from one device which constructs a list of all other Bluetooth devices in the neighborhood that are publicly discoverable and asks the user to choose the right one to pair with. In the out-of-band association model, device addresses are sent via the out-of-band channel. This makes it possible to uniquely identify the peer to pair with, without requiring user selection. In the other association models, SSP does not contain any new mechanisms to make peer discovery easier. Individual implementations could use existing Bluetooth modes, like the 'limited discoverable mode' and 'pairable mode' to support user conditioning on the peer device. In user conditioning, user sets conditions (e.g. a time period) to control how and when a device can be paired. However, since such user conditioning is not mandated by the specification, it is quite possible that the implementations of SSP may still need to resort to asking the user to choose the right peer device from a list.

Model selection: The association model to be used is uniquely selected during the initialization of the session. If the association process is initiated by out-of-band interaction, and security-information is sent through the out-of-band channel, then the out-of-band model is chosen automatically. Otherwise, in phase 1, the devices exchange

their input-output capabilities. The SSP specification describes how these capabilities should be used to select the association model.

2.1.4.2 Wi-Fi Protected Setup

Wi-Fi Protected Setup (WPS) is Wi-Fi Alliance's specification for secure association of wireless LAN devices. Microsoft's Windows Connect Now (WCN) includes a subset of association models described in WPS. The objective of WPS is to mutually authenticate the enrolling device with the Wi-Fi network and to deliver network access keys to the enrolling device. This is done by having the enrolling device interact with a device known as the ``registrar'', responsible for controlling the Wi-Fi network. The registrar may be, but does not have to be, located in the Wi-Fi access point itself. WPS supports three configuration methods: In-band, out-of-band, and push-button configurations.

In-band configuration enables associations based on a shared secret passkey (Figure 4: P7). The user is required to enter a passkey of enrollee to the registrar. This passkey may be temporary (and displayed by the enrollee) or static (and printed on a label). 8-digit passkeys are recommended but 4-digit passkeys are allowed. The passkey is used to authenticate the Diffie-Hellman key agreement between the enrollee and the registrar. The protocol used is a variation of the modified MANA III protocol in Figure 6 with two rounds (k=2). As in MANA III, once a passkey is used in a protocol run, an attacker can recover the passkey by dictionary attack (although in this instantiation, the attacker needs to be active since the computation of the used commitments includes a key derived from the Diffie-Hellman key).

1. **Out-of-band configuration** is intended to be used with channels like USB-flash drives, NFC-tokens or two-way NFC interfaces. There are three different scenarios:

    - Exchange of public key commitments (**Figure 4**: **P4**), typically intended for two-way NFC interfaces, where the entire Diffie-Hellman exchange and the delivery of access keys takes place over the out-of-band channel.

- Unencrypted key transfer (**Figure 4**: **P1**). An access key is transmitted from a registrar to enrollees in unencrypted form, either using USB-flash drives or NFC-tokens.

- Encrypted key transfer. This is similar to the previous case, except that the key is encrypted using a key derived from the (unauthenticated) Diffie-Hellman key agreed in-band. From a security perspective, this is essentially out-of-band key transfer (**Figure 4**: **P1**).

2. **Push button configuration** is an optional method that provides an unauthenticated key exchange (**Figure 4**: **P11**). The user initiates the Push button configuration by conditioning the enrollee (e.g., by pushing a button), and then, within 120 seconds the user has to condition the registrar as well. The enrollee will start sending out probe requests to all visible access points inquiring if they are enabled for push button configuration. Access points are supposed to respond affirmatively only when their registrar has been conditioned by the user for this configuration. If a device or registrar sees multiple peers ready to start push button method, it is required to abort the process and inform the user.

Peer discovery: Enrollees start association in response to explicit user conditioning. They scan the neighborhood for available access points and send Probe Request messages. The Probe Response message has a ``SelectedRegistrar" flag to indicate if the user has recently conditioned a registrar of that access point to accept registrations. This is mandatory for push button configuration but is optional for other models. Thus it is possible that user may have to be asked to select the correct Wi-Fi network from a list of available networks.

Model selection: The model is explicitly negotiated at the beginning of pairing between the paired devices.

2.1.4.3 Wireless USB Association Models

Wireless USB (WUSB) is a short-range wireless communication technology for high speed data transmission. WUSB Association Models Supplement 1.0 specification

from [52] supported two association models (cable and numeric) for creating trust relationships between WUSB hosts and devices. The new specification [39] supports three models:

1. **Out-of-band model** uses OOB key transfer (**Figure 4**: **P1**) and utilizes e.g. wired USB connection, NFC or memory cards to associate devices. Connecting two WUSB gadgets together is considered as an implicit decision and, hence, the standard does not require users to perform additional actions like accept user prompts.

2. **Fixed symmetric key association** model relies on authenticated symmetric crypto key agreement (**Figure 4**: **P2**). End user provides USB device's symmetric key to the USB host device. Device can then connect to host in order to prove that both devices know the symmetric key and to agree on device specific secret AES key.

Numeric model (In-band key exchange) relies on the users to authenticate the Diffie-Hellman key agreement by comparing short integrity checksum values (**Figure 4**: **P5**). The protocol is an instantiation of the protocol in Figure 5. First D_A and D_B negotiate the length of the checksum to be used. The specification requires that WUSB hosts must support 4-digit checksums whereas WUSB devices must support either 2 or 4-digit checksums.

Peer discovery: The association is initialized by implicit or explicit user conditioning. Attaching a USB-cable is interpreted as an implicit conditioning. The user pressing a button is an example of explicit user conditioning. In the numeric model the user sets a USB device to search for hosts and a USB host to accept connections. The host advertises its willingness to accept a new association in the control messages it transmits on the WUSB control channel. In case multiple devices are simultaneously advertising their accepting states, the searching device either selects a host randomly or ends the association procedure in a failure.

Model selection: The choice of the association model is based on the type of user conditioning done. In case a cable is plugged, the devices exchange information on

whether they support OOB association. If so, they use OOB model. If conditioning is explicit, they use numeric model or symmetric key depending of the device's capabilities.

2.1.4.4 HomePlugAV Protection Modes

HomePlugAV is a power-line communication standard for broadband data transmission inside home and building networks. Typically, several apartments share a power-line network. In addition to protecting deliberate attacks, association mechanisms are used to create logically separate subnetworks by distributing a 128-bit AES network encryption key (NEK) for devices in each subnetwork. As with WPS, each HomePlugAV network has a controller device. HomePlugAV supports the following association models [40]:

1. **Simple connect mode** uses symmetric crypto based key agreement to agree on a shared key. This network membership key (NMK), is used to transport NEK to the new device. The key agreement process is as follows. To admit a new device, the user is required to first condition the controller device, and then condition the new device, e.g., by turning on its power. The devices find each other and exchange nonces. A temporary encryption key (TEK) is formed by hashing the two nonces together. The controller encrypts the NMK using the TEK and sends it to the new device. The model is unauthenticated (**Figure 4**: **P3**) as no cryptographic authentication mechanisms are used.

2. **Secure mode** allows new devices to have a secret passkey, of at least 12 alphanumeric characters long, typically printed on a label. The user is required to type in this passkey to the controller device. This is an example of authenticated symmetric crypto key agreement (**Figure 4**: **P2**). The controller device uses passkey to construct an encryption of NMK and send it to the new device. The keys for devices joining in secure mode are different from the keys for devices joining in simple connect mode.

3. **Optional modes** enable use of alternative models for distributing NMKs or NEKs between devices. These include ``manufacturer keying" where a group of devices have a factory installed shared secret, and external keying, where trust is bootstrapped from other methods.

Man-in-the-middle attacks can be prevented in simple connect mode by utilizing characteristics of powerline medium. Before two nodes can communicate, they must negotiate tone maps, which enable devices to compensate disturbances caused by powerline channel. This negotiation is done in a reliable, narrow-band broadcast channel. Thus a man-in-the-middle trying to negotiate tone maps with the legitimate endpoints can be detected.

Passive eavesdropping in the broadband point-to-point channel is difficult since an attacker, even with the knowledge of the tone maps used between the legitimate endpoints, will not be able to extract the signal from the channel because the signal-to-noise ratio will be too poor at different locations, particularly, when the attacker is outside a building and the legitimate end points are inside. Also, licensees of HomePlugAV technology do not provide devices that can extract signal without negotiating tone maps. Hence, attackers must be able to build expensive devices for eavesdropping.

Peer discovery: In simple connect mode the peer discovery is performed by the user conditioning the devices into a suitable modes, and the new device scanning the network to find a controller that is willing to accept new devices.

Model selection: The model is selected by user conditioning. There is no automatic negotiation.

2.1.5 Security Evaluation and Analysis

In this section, we analyze the association models described in the previous subsection from different perspectives and point out some problematic areas.

2.1.5.1 Comparison of Security Levels

First we summarize and compare the security levels provided by the different key establishment protocols. A comparative summary of models' security characteristics is presented in Table 2.

Table 2. Comparison of security characteristics of key establishment models in different standards [Article I]

Association model	*Offline attacks*		*Online active attacks (MitM)*		
	Protection	*Work*	*Protection*	*Success probability*	*Work*
Bluetooth Secure Simple Pairing					
Numeric Comparison	DH	2^{80}	6 digit checksum	2^{-20}	2^{148}
Just Works	DH	2^{80}	-	1	0
Passkey Entry	DH	2^{80}	6 digit checksum	2^{-19}	2^{147}
Out-of-band	DH	2^{80}	OOB security	-	2^{128}
Wi-Fi Protected Setup					
In-band	DH	2^{90}	8 digit checksum	$2^{-13.2}$	$2^{141.2}$
In-band + OOB	DH	2^{90}	OOB security	2^{-128}	2^{196}
OOB	OOB	2^{90}	OOB security	-	-
PushButton	DH	2^{90}	-	1	0
WUSB Association Models					
Numeric Model	DH	2^{128}	2/4 digit checksum	$2^{-6.6}$ or $2^{-13.2}$	$2^{262.6}$ or $2^{269.2}$
OOB model	OOB	2^{128}	OOB		
HomePlugAV Protection Modes					
Simple Connect	SNR	High	traffic monitoring	low	High
Secure Mode	AES	2^{72}	passkey	2^{-72}	2^{72}

2.1.5.1.1 Offline Attacks

The out-of-band association models rely on the secrecy of out-of-band communication to protect against passive attacks against key agreement. The in-band and hybrid models in all of the standards except HomePlugAV use Diffie-Hellman key agreement to protect against passive attacks. The level of protection depends on the strength of the algorithms and the length of the keys used. In the 'Work' subcolumn under the 'Offline Attacks' column of Table 2, we use [53, 54], to estimate the amount of work an attacker has to do in order to be successful. The figures correspond to approximate lower bounds, and should be treated as rough estimates only. Offline attack protection in HomePlugAV relies on the characteristics of the power-line communications: the proposal [40] assumes that signal-to-noise ratio (SNR) makes it difficult for an attacker to eavesdrop. The HomePlugAV secure mode uses symmetric key encryption as protection.

2.1.5.1.2 Online Active Attacks

In online active attacks, a *man-in-the-middle* attacker must be able to intercept transmissions and modify it without causing delays or disturbances, which will cause attack to be detected. Hence, several of the models (Bluetooth Just Works, Wi-Fi Push Button, and HomePlugAV Simple Connect) trade off protection against man-in-the-middle attacks, in return for increased ease-of-use.

Other in-band association models rely on authentication as the means to protect against online active attacks. The probability of success for an online active attack depends on the length of the key as well as the protocol. The Bluetooth SSP numeric comparison model uses 6-digit checksums leading to a success probability of 1/1000000. The WUSB numeric model allows a success probability of 1/100 when two digit checksum is used, and 1/10000 when four digit checksum is used. These probabilities do not rely on any assumptions about the computational capabilities of the man-in-the-middle.

Association models based on numeric comparison use cryptographic hash functions as the commitment function. In principle, a man-in-the-middle, who can break the hiding property of the hash commitment function during the key agreement process, can also succeed by figuring out the nonce used in the commitment. 'Online Active Attacks – Work' column in Table 2 shows the amount of on-line work (exhaustive search) the attacker has to perform in order to succeed with probability 1. If the hash function is strong, and requires exhaustive search to find the correct pre-image, the work factor depends on the size of the nonce and the size of the checksum. Bluetooth SSP uses 128-bit nonces and 20-bit checksum; therefore, the attacker must make 2^{148} quesses. WUSB numeric model uses the Diffie-Hellman public value as the hidden nonce, which is based on a 256-bit long private value. It uses 2- or 4-digit checksums. Hence, work factor figures of $2^{262.6}$ or $2^{269.2}$ are used. These figures correspond to the amount of on-line work required for the attacker to succeed with probability 1.

Association models based on passkeys also use cryptographic hash functions as the commitment function. An attacker who can break the hiding property of the hash function can figure out the nonce and the passkey component used in a given round. The work factor depends on the size of the nonce plus the size of the passkey component. For Bluetooth SSP the work factor is 2^{147} (128-bit nonce and 19-bit passkey

component), whereas for WPS in-band model the work factor is $2^{141.2}$ (128-bit nonce and 4-digit passkey component). Alternatively, an attacker who can break the binding property of the hash function can send a randomly chosen value as h_{i2} in Step 2 of the protocol in Figure 6, learn the passkey after receiving message 3 and then calculate a suitable R_{i2} that matches the alleged commitment sent earlier in Step 2. The work factor depends on the size of the commitment. Bluetooth SSP uses 128-bit commitments, leading to a work factor of 2^{128}. WPS uses 256-bit commitments leading to larger work factor for breaking the binding property than breaking the hiding property. Therefore, the $2^{141.2}$ work factor needed for breaking the hiding property is used.

Recall from Subsection 2.1.2 that with n bit passkeys and k rounds the success probability for an online active attack against the passkey protocols is $2^{-(n-(n/k))}$. Bluetooth SSP passkey entry model uses 6-digit ($n \approx 20$) one-time passwords in k=20 rounds. This leads to approximately 1/1000000 success probability. WPS network uses essentially the same protocol, but in two rounds only. This leads to success probabilities of 1/100 when 4-digit passkeys are used, and 1/10000 when 8-digit passkeys are used. In both cases, the passkey must be single-use. If the passkey is re-used, the success probability of man-in-the-middle rises dramatically, reaching 1 after the k^{th} re-use, where k is the number of rounds in the original protocol. In other words, if the same fixed passkey in WPS network model is re-used even once, the man-in-the-middle can succeed in the next attempt with certainty. As before, we can estimate the on-line work effort the attacker has to do to break the hash commitments.

HomePlugAV secure mode uses a 12 character passkey which is used to generate a key for AES encryption, leading to a probability of 2^{-72} and the amount of on-line work effort is 2^{72}. Attack probability against HomePlugAV simple connect mode is assumed to be small as attackers can be detected by monitoring communication on narrowband channel [40]. However, the security level has not been formally proven.

In the Wi-Fi hybrid model, the random secret, transferred through one-directional out-of-band channel, is 128 bits long leading to a computational security of 2^{-128}.

2.1.5.2 Associations with Wrong Peers

Unauthenticated association models face the risk of a device being associated with a wrong peer. For instance, in WPS push button model, the user may condition first the enrollee to search for registrars before conditioning the registrar. If the attacker sets a bogus registrar to accept connections before the users does it with the legitimate registrar, the enrollee associates with the attacker's registrar. Only in the case when both registrars, the bogus and the legitimate one, are simultaneously accepting connections, is the procedure aborted.

In HomePlugAV Simple Connect mode, the user sets the control device to accept connections before starting the joining device up. This could be used to reduce the probability for an attacker to successfully masquerading as a bogus control device because since, if the new device sees multiple control points, it can abort association. However, the mode is potentially vulnerable for fatal errors where the user is slow to switch power to the new device. In this case an attacker may connect to user's control point and get the network encryption key. The longer walking distance there is between power-line devices, the more likely this attack is to succeed.

2.1.6 Challenges with Devices Implementing Multiple Key Establishment Models

The previous section presented straightforward attacks against individual key establishment models and how naive implementations of user interaction could increase the likelihood of fatal errors. This section presents novel attacks arising out of the fact that the standards invariably support multiple association models simultaneously.

Consider specifications that support an unauthenticated association model as well as user-assisted comparison of integrity checksums. An example is a Bluetooth device that supports the numeric association model and the unauthenticated ‘Just Works’ model. Figure 8 illustrates a man-in-the-middle attacker who can intercept messages exchanged during an association. The first associated device has a display and the second may or may not have a display. The attacker changes device capability information so that the first device will be using the numeric comparison model and that the second device will be using unauthenticated ‘Just Works’ model. This leads to a

situation where the first device shows a 6-digit checksum and the second device, using 'Just Works' model, does not display a checksum, even if it would have a display. The user may have been educated to detect a mismatch in checksums. But now, when only one device displays a checksum, the user is likely to be confused and may just go ahead and accept the association.

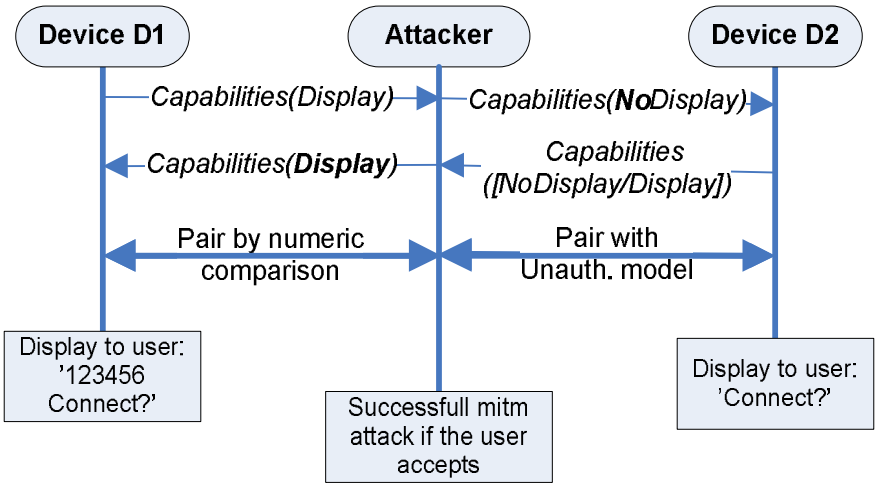

Figure 8. Man-in-the-middle between different association models [Article I]

To get an idea about whether such user confusion is likely, a laboratory usabilility test study, presented in [55], tested the attack. Out of 40 test users, 6 accepted the pairing on both devices, 11 noticed the problem and rejected the pairing on both devices, and the rest rejected pairing on Device 1 but accepted it on Device 2.

This attack has two implications.

1. When the second device has a display, it is a *bidding down attack* against this device. The second device will know that the association is unauthenticated. However, the user may still allow the association to happen.

2. It is *a bidding up attack* against the first device since it 'believes' that the association is made using a secure protocol resistant to man-in-the-middle attacks. Consequently, the first device may choose to trust this security association more than it would trust a 'Just Works' security association. For instance, it may have a policy rule, which allows more trustworthy devices to initiate connections without run-time confirmation from the user.

A scenario related to the attack on the figure arises with devices that are willing to participate in setting up a security association without immediate user conditioning. Public printers and access points are examples of devices that may be permanently conditioned for association. Suppose a user starts associating Device 1 with Device 2 using an association model that does not require any user dialog (e.g., WUSB cable model, or HomePlugAV Simple Connect mode) and that Device 2 is permanently conditioned to accept incoming association requests. If an attacker now initiates association with Device 2, say using Bluetooth SSP numeric comparison, a user dialog will pop up on Device 2. Since the user is in the middle of associating Device 1 and Device 2, he might answer the dialog thinking that it is a query about Device 1. Depending on the nature of the dialog, the attacker may end up gaining unintended privileges on Device 2.

Wi-Fi and Bluetooth have legacy association models. They use symmetric algorithms with pre-shared key or personal identification number (PIN). If a device supports both the improved and the legacy association models, it is vulnerable to bidding down attack. This attack is difficult to detect as the user is required to be aware that both devices support particular association models and then enforces that this models is actually used.

2.1.6.1 Strengthening Devices

The attacks against standardized mechanisms, identified above, can be addressed with implementation decisions. When a security association is stored persistently, information about its level of security should be stored as well. HomePlugAV already does this indirectly by using different keys with different association models. Furthermore, this security-level information should be used in deciding what the peer device is authorized to do. For instance, devices associated using Bluetooth SSP `Just Works' or HomePlugAV Simple Connect models should not be allowed to install or configure software, at least, without explicit authorization from the user. This precaution would help to mitigate the consequences of bidding down attacks. The man-in-the-middle attack between numeric comparison and unauthenticated protocols (Figure 8) could be addressed with two alternative strategies:

1. Bidding down the second device from using numeric comparison to the 'Just Works' model could be addressed by requiring that devices believing to be in 'Just Works' association would anyway show the checksum if they are able to do so. However, this solution does not prevent the bidding up attack against the first device.

2. Bidding down and bidding up attacks can both be countered by querying the user appropriately to confirm the I/O capabilities of the peer device. For instance, if the capability negotiation messages indicate that the peer device has no display, a device could ask the user if the peer device does indeed have a display. If the user gives answers affirmatively, it is an indication of a man-in-the-middle. However, such an additional dialogue is likely to have negative effects to usability.

2.2 A Mediator for Key Establishment

The protocols presented in the previous subsection enable secret keys to be established for different kinds of devices. However, in practice end-users may end up to a situation were devices have incompatible physical interfaces making secure key establishment impossible. This subsection addresses these concerns and studies how the mediator concept can be utilized to solve these interoperability problems. Particularly, this subsection contributes by proposing protocols for mediating pairing for cases where devices have different types of secure interfaces available. The contributed protocols and mechanisms are targeted for extending devices, which support the Bluetooth Secure Simple Pairing standard, to support mediators. Subsection 2.2.1 defines the problem. Subsection 2.2.2 proposes pairing protocols for different types of OOB channels and also presents challenges and questions for further studies needed clarification before the mediator concept can be realized in universal manner.

2.2.1 The Interoperability Challenge Caused by Use of Diverse Key Establishment Mechanisms

The emerging key establishment mechanisms and association models provide several usability, cost and security advantages but also cause new challenges. Support for

multiple mechanisms will cause interoperability problems since, even if some devices may support many mechanisms, every device do not support every available option. Two devices, which do not have compatible interfaces for key establishment, cannot be associated. This problem emerges when several different association models and out-of-band interfaces are adopted to personal devices. For instance, the new Bluetooth Secure Simple Pairing standard, which we presented in Subsection 2.1.4.1, enables devices to be paired using OOB channels like NFC or by comparing values displayed by paired devices. If one device has only NFC interface but no display or keyboard and another device has only display and keyboard but no NFC interface, these devices cannot be paired securely. Alternatively, if both devices have only low-cost passive NFC tags, they cannot be paired securely.

2.2.2 A Mediator for Bluetooth Secure Simple Pairing

The interoperability challenge, of pairing devices with incompatible pairing interfaces, can be addressed with a mediator concept. Mediators are devices, which support several association interfaces. For instance, a mediator could be mobile phone, tablet, or personal computer etc. The mediator is a trusted device, which must be available during the pairing process. After the pairing, associated devices can communicate directly with each other without the help of the mediator.

Mediating has been used in key establishment in several occasions. For instance, Touch mediated Association Protocol (TAP) [56] is a solution where the end-user touches two devices with a third one in order to pair them. Tapping is based on transmitting secrets through a short range wireless channel. The solution assumes that both paired devices support this channel. Also, WLAN access points can in some sense be considered as mediating devices. However, WLAN security methods control only which devices can join a network. They do not provide fine-grained control over device to device communication.

Figure 9 illustrates generic components needed in the scheme. In the figure, components, which are new and must be added to enable use of mediators, are emphasized with darker blue. The secure channel, in the figure, may be either an authenticated Bluetooth channel or an OOB channel depending of the scenario.

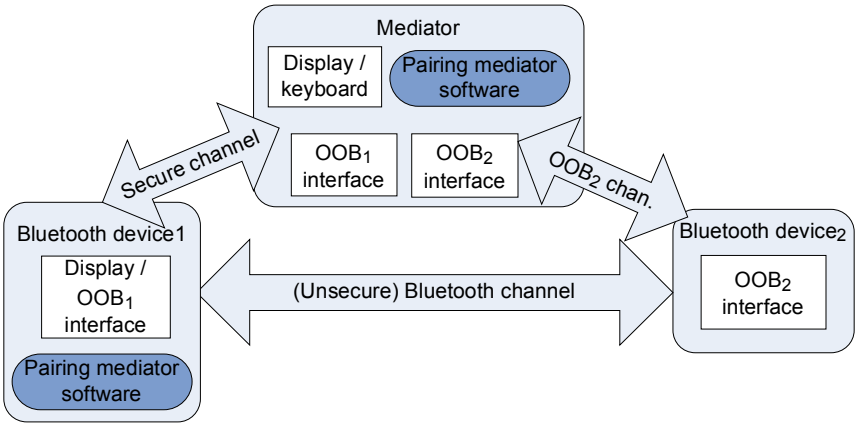

Figure 9. Components in the mediator-based pairing

The main motivation to mediator comes from the interoperability. Use cases for mediator include e.g. a pairing between a television (with a display and Bluetooth) and cheap speakers (with NFC interface and Bluetooth). Clearly, these devices could not be paired securely (without the risk of active man-in-the-middle attack) if there would not be a mediator device. However, the interoperability is not the only reason to use mediators. Two devices may have compatible OOB interfaces but it still may be more usable to use mediator instead of direct OOB connection. For instance, consider a case where there are NFC and Bluetooth enabled television and NFC and Bluetooth enabled air-conditioning system, which starts to blow when a storm is displayed in television. These devices may be located so that they cannot be connected with NFC. Due to their weight, they cannot be moved and associated. In these cases, a mobile phone acting as a mediator provides an easy alternative for making the pairing.

The mediator-based pairing enables manufacturing of devices with lower costs and lower power consumption. This is because it is possible to select cheaper hardware interfaces alternatives to devices. Manufacturers do not have to implement expensive two-directional interfaces to their devices. For instance, it is enough that devices have passive NFC tags, which can only send data. Alternatively, when considering associations through optical channel, devices can have only cheap transmitters like LEDs. Only the mediator devices must have more capable interfaces like NFC reader or camera.

The mediation between two OOB devices scenario may have a positive side effect to the usability. Firstly, as emphasized in TAP [56], a mediator provides natural way to show which devices must be paired. There is no need for an additional (e.g. broadcast based) device discovery or selecting devices from a long list in UI. Secondly, a mediator can provide easy interface to manage pairings between devices which itself do not have displays or have smaller displays. The mediator provides also always a consistent user interface as well as dialogs and thus minimizes fatal user errors. Thirdly, a mediator does not have to make the transfer of pairing information at once. It may be used to store this information potentially for very long periods of time. For instance, when a new device is brought to home, the user pairs it with the mediator. Consequently, the new device will receive information from every device (and make pairing with these devices), which has provided its information to mediator. Also, the device will leave its pairing info to the mediator so that newer devices will also be able to make the association.

2.2.2.1 Mediator-based Association Models

The mediator-based pairing may be initiated in different ways. The user may utilize either mediator device's user interface (display) to select devices to be paired. Alternatively, the end-user may use user interface of Device *D1* to search other devices to be paired. Also, the user may perform pairing simply by first touching a mediator device with one device and then with another.

The end-user can trigger secure pairing and select paired devices by using a mediator, which scans Bluetooth network and displays identities of pairable devices (the user must have conditioned paired devices so that they are visible e.g. they have a special button for this). The user then selects those which should be paired. After this the user is asked to first pair another device with a mediator and secondly another device with a mediator e.g. by touching a new device with a mediator.

The second alternative is that the user uses Device *D1*, which has a display, to search for a pairable device. If this device detects devices with incompatible pairing interface, it scans network using Bluetooth service discovery mechanism to detect a mediator service, which would support mediation between these devices.

In the third alternative, the end-user indicates devices to be paired in a physical manner for instance by moving a device to close of mediator or by connecting mediator to device with a cable. Additionally, the user may have to condition the mediator and devices into a mode where pairing may happen.

2.2.2.2 Pairing Protocol for Direct Mediator-based Association

The protocols for pairing depend on the hardware capabilities of the paired devices. In this subsection, we will consider three cases, which are different in a sense that the directionality of channels is different.

The simplest mediator-based pairing case is when the one device is able to send data through a secure (e.g. OOB) channel and another device is able to receive data through a secure channel.

The protocol for this case is described in Figure 10. Mediator's role can be considered to be a one directional OOB channel where a mediator is used to forward a secret and a commitment. Afterwards devices can finalize the pairing through the unsecured Bluetooth channel. In the figure, messages which are transmitted though out-of-band channel are illustrated with arrows in rectangles.

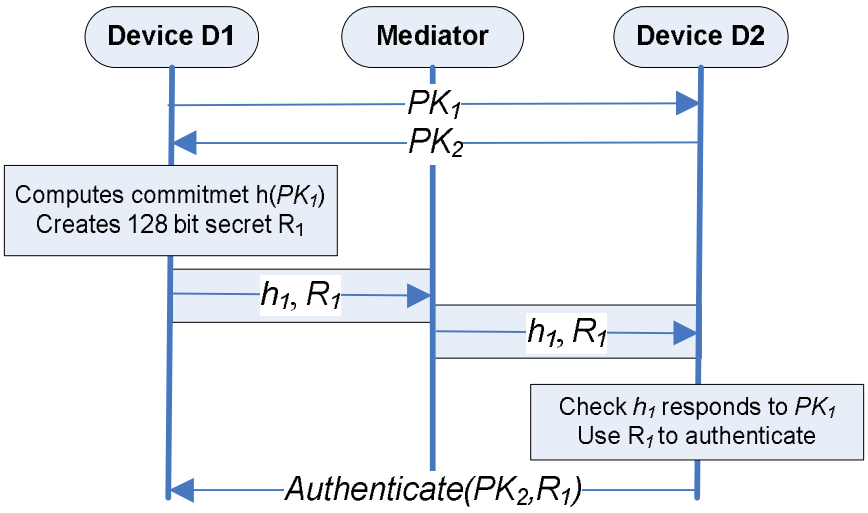

Figure 10. A protocol for pairing a device, which has an outbound secure channel, and a device, which has an inbound secure channel

The protocol has the following steps:

1. Devices change public keys PK through BT channel

2. D1 computes commitment to public keys PK $h_1 = h(PK_1)$ and sends it and 128-bit secret R_1 to M (The end-user may be required to make a pairing between M and D1 before this can be done securely)
3. M forwards commitment and secret to D2 using OOB channel
4. Normal Secure Simple Pairing Protocol continues (D2 checks that commitment h_1 responds to public key PK_1 and uses secret R_1 to authenticate itself to D1)

Devices to be paired perform input/output capability exchange. In Simple Pairing, this enables devices to select correct association algorithm. However, when interfaces are incompatible as in our case, this may cause the pairing process to stop. Consequently, Device *D1*, which sends commitment and secret, must receive information that Device *D2* has compatible I/O capabilities. This can be achieved in two ways: either Device *D2* is compatible with our protocol and, hence, able to advertise that it has compatible interface even if it does not have, or the mediator is able to intercept and modify capability negotiation messages.

2.2.2.3 Pairing Protocol for Devices with Only Outbound OOB Interfaces

It may not be always possible that a secret and a commitment can be transmitted from one device to mediator and received in another from mediator. Instead, both devices may be only able to send OOB data. For instance, both devices may have passive NFC tags but no NFC readers. A mediator-based protocol for pairing two devices, which can only send association data is presented in Figure 11. In the figure secure messages, which are secured using established Bluetooth secure connection, are illustrated with bolder arrows.

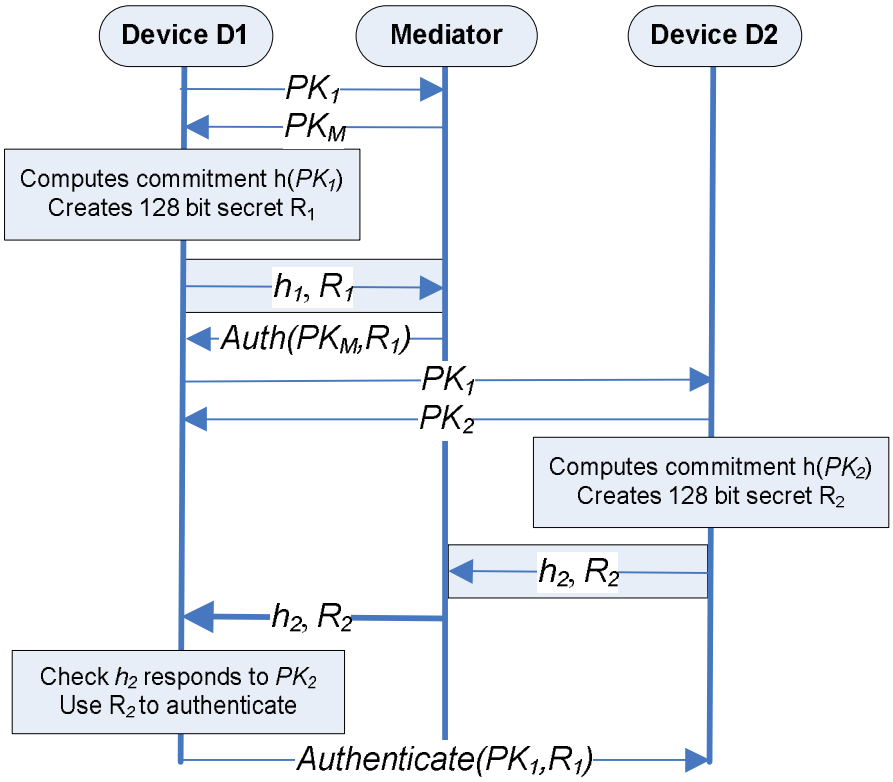

Figure 11. A protocol for pairing two devices, which have only outbound OOB channels

The protocol has the following steps:

1. Mediator M changes public keys *PK* through BT channel with D_1
2. Device D_1 computes a commitment to public key *PK* $h_1 = h(PK_1)$ and sends it and 128-bit secret R_1 to M via OOB channel
3. M uses commitments, public keys and secrets to make Secure Simple Pairing with D_1
4. Devices D_1 and D_2 change public keys *PK* through BT channel
5. D2 computes a commitment to public key *PK* $h_2 = h(PK_2)$ and sends it and 128-bit secret R_2 to M via OOB channel
6. M forwards secret and commitment to D1 using secure channel
7. Normal Secure Simple Pairing Protocol continues (D_1 checks that commitment h_2 responds to public key PK_2 and uses secret R_2 to authenticate itself to D_2)

2.2.2.4 Pairing Protocol for Devices with Only Inbound OOB Interfaces

If both devices can only receive secure data, a mediator must first be paired with both devices. The pairings between a mediator and devices is created so that mediator sends secure association data to both devices to create pairings between them. The mediator

then transmits association information through secure channels, which it has established with both pairable devices. For instance, a mediator with an NFC reader may be paired with two devices with NFC transmitters. Then, this mediator may transport association information from one device to another. The solution requires both devices to support pairing through a mediator. The protocol is illustrated in Figure 12.

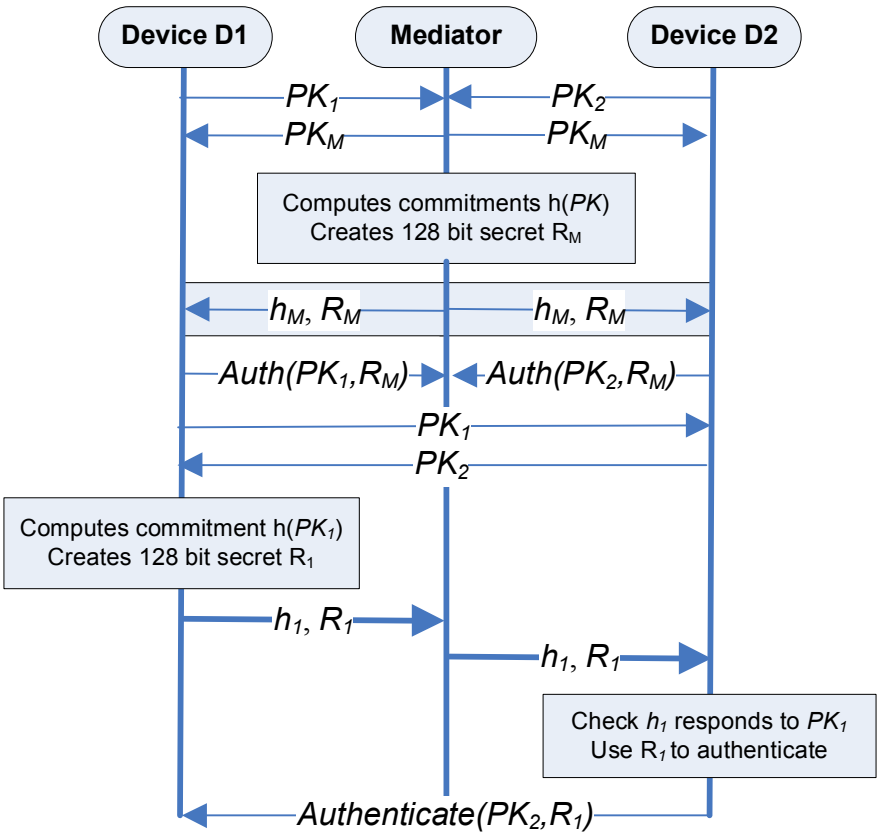

Figure 12. A protocol for pairing two devices, which have only inbound OOB channels

The protocol has the following steps:

1. Mediator (M) changes public keys *PK* through BT channel with both device
2. M computes commitments to public keys *PK: h* = h(PK) and sends them and 128-bit secret R_M to devices via OOB channel
3. Devices use commitments, public keys and secrets to make Secure Simple pairings with M
4. Devices D1 and D2 change public keys *PK* through BT channel
5. D1 computes a commitment to public key PK h_1 = h(PK_1) and sends it and 128-bit secret R_1 to M
6. M forwards secret and commitment to D2

7. Normal Secure Simple Pairing Protocol continues (D2 cheks that commitment h1 responds to public key PK1 and uses secret R1 to authenticate itself to D1)

2.2.2.5 Towards a Universal Mediator

The described protocols are specific for the Bluetooth Secure Simple Pairing standard. Ideally, mediator-based pairing could be utilized also with other devices with incompatible security association interfaces. However, in practice achieving a universal mediator solution, which would solve all pairing related problems in the connectivity layer, is challenging:

- The proposed model is suitable only for devices supporting particular protocols and association models. However, for instance, in Bluetooth Secure Simple Pairing, I/O capability negotiation occurs directly between devices. If the devices believe to have incompatible interfaces they do not continue the pairing procedure. Consequently, at least one (depending of the directionality of OOB channels) device must be compatible with the scheme and able to advertise appropriate capabilities. The standards, which support versatile association models, could also include support for mediator based association. However, achieving standard level interoperability is not an easy task.

- It is difficult to maintain the control over security levels. When using a mediator, an associated peer believes that the peer has mediator's security capabilities and, hence, may give the peer device undeserved privileges.

- Devices may support various connectivity mechanisms; say Bluetooth and Wi-Fi. These protocols are incompatible and have e.g. different security concepts and credential formats. Consequently, the user is ofter required to assist in security pairing several times by using different user interfaces.

Hence, this thesis promotes the idea that higher-level solutions are needed for establishing secure associations between heterogenous devices. The forthcoming sections will present security platforms and middleware solutions. In Section 5.2, controlled and connectivity-independent key establishment is achieved with a brokered middleware solution.

3 Certification and Reputation based Security

The previous section focused on technologies where the end-user controls devices and introduces devices to each other. However, there are large amount of network applications where devices are not in the control of the user. Instead, the user must identify and trust devices belonging to others. The security model of the Internet is based on the trusted third-parties, who provide certifications and security reputation information. This security model is scalable and suitable for heterogenous services, as clients can authenticate servers and resolve relevant security information using a uniformly presented data structures. This section surveys these security solutions. The section focuses on SSL/TLS protocols, certification, and reputation management. Particular focus is on possibility to use these mechanisms to provide more detailed and rich information, which can be used in smart authorization solutions.

The section contributes by providing a large-scale empirical analysis on the correlation of SSL certification and crowd-based reputation evaluations. The study, first presented in Article VII, has two implications. Firstly, it introduces a novel metric that can be used when analysing impacts and visibility of web security solutions. Secondly, correlation information is used to get some indications of the benefits that web services gain from SSL certification, extended validation, and selection of more reputable certification authorities.

3.1 SSL/TLS

Secure Socket Layer (SSL) and, IETF's standardized version, Transmission Layer Security (TLS) [57] protocol have been designed to secure and mutually authenticate applications on top of the Internet Protocol (IP). However, the protocol can be used also on top of other protocols. TLS has been widely accepted defacto standard for different IP based applications starting from WWW. It provides a scalable and flexible mechanism by supporting various security algorithms. There are several protocol stack implementations available and the protocol is mature and high-secure.

Key establishment of TLS falls into authenticated exchange of commitments via unspoofable channel category (**Figure 4:**P4**)**. The key commitments are certificates and the unspoofable channel is public key infrastructure. Trusted third parties (certification authorities) verify identities of public key holders and sign matching public keys. The signing is done using root signing keys that other devices can verify to belong to trusted authorities by checking them against root certificates installed to these devices. Devices connecting to each other use certificates to negotiate a session key during a three way handshake. The TLS protocol supports different asymmetric algorithms for key establishment handshake and also various symmetric crypto algorithms for securing communication session.

Feasibility and costs caused by TLS in Internet applications has been analyzed in several papers including [58, 59]. Experiences of TLS's performance indicate that the main penalty is related to the handshake phase. For instance, Du et al. described [60] measurements for their SNMP implementation where latency for the first TLS secured packet in a session was over 8 times larger than the first unsecured UDP packet and 12.5 times larger than the later TLS packets in the same session. In the following packets, the performance penalty was only around ten to twenty percent.

3.2 SSL Certification in WWW

Authentication and confidentiality of communication in the World Wide Web (WWW) is based on HTTPS (Hypertext Transfer Protocol Secure) [61], where communication is protected with SSL (Secure Sockets Layer) [57] protocols, as well as X.509 public key certificates [62, 63], which vouch the identities of services. The authentication model is scalable and capable of preventing most masquerading attacks when used properly. The model has, however, been criticized due to large amount of equally trusted certification authorities (CAs) and loose certification processes, which make acquiring of phishing certificates possible for attackers. Extended validation [64] certificates and additional visual trust indicators in browsers have been proposed as a more secure certification alternative. However, there have not been large scale studies on the benefits that the service providers gain from SSL certification in general and from extended validation.

Authentication of web servers is based on X.509 certificates, which have been granted to servers by a trusted CA. In typical browsers (including Mozilla Firefox, Internet Explorer, Google Chrome etc.) the amount of accepted root certificates is large. The acceptance criteria depend on the trustworthiness of CA but also on business and politics. If one of these CAs has been compromised and certifies bogus servers, the end-users' web transactions are in jeopardy. Browser's security identifiers will not warn on bogus servers certified by trusted CA even if it would have been a different CA that actually had signed the victim service. Attacks demonstrating the weaknesses of CAs have already been reported, including the recent DigiNotar and Comodo incidents [65, 66].

Large scale studies on how the certificates are used has been performed by Eckersley et al. [67], who scanned public Internet for certificates and reported several vulnerabilities. Vratonjic et al. [68] analyzed certificates with the million most popular web sites and reported that most HTTPS servers do not use certificates properly. Typical problems are domain mismatch, certificate expiration and untrusted (self-signed) certificates.

Dhamija et al. [69] studied users' ability to distinguish real web sites from spoofed sites using SSL warnings. They found that 23% of participants did not check browser's passive security indicators at all when evaluating the trustworthiness of the site. Sunshine et al. [70] performed a survey and a laboratory test to examine users' reactions to different active SSL warnings. They noted that users' behaviour depends on the actual message as well as on the service type. Tests revealed that more than the half of the hundred participants ignored the warnings of the main stream browsers and proceeded to the web sites anyhow. A bit more moderate results were gained by Egelman et al. [71] who found that 21% of sixty study participants ignored active warnings and fell to phishing attacks. When the security indicators and warnings are ignored, the credibility of a web site depends on various other factors. These factors were studied by Fogg et al. [72]. Their study, made with 1400 participants, reveals that real-world feel, ease of use and expertise are the most important categories affecting to credibility.

SSL certificates are assigned to service providers through diverse certification processes. Typically, it is enough that the requester has an access to email, which has been registered for the domain name holder. This makes acquirement of phishing certificates possible for attackers. Some certification authorities may have more trustworthy processes in use but the large amount of equally trusted authorities means that end-users do not have practical means to separate real and trustworthy certifications from bogus certification received from a compromised or careless authority.

3.2.1.1 Extended Validation

Extended Validation Certificates [64] and additional visual trust indicators in browsers have been proposed as a more secure certification alternative. EV certificates are given for servers, which have gone through stricter authentication processes. Browsers identify servers with EV certificates as more trusted by displaying additional trust indicators, notably green address bar. See Figure 13 and Figure 14 for examples of address bar in Mozilla Firefox 8 and Internet Explorer 8 looks when browser connects to services with either unsecure HTTP, (ignored) invalid certificate on HTTPS server, valid regular certificate on HTTPS server, or EV certificate on HTTPS server. EV trust indicators have been supported for a couple of years in the main stream browsers including Microsoft Internet Explorer (since version 7, released October 2006), Mozilla Firefox (version 3, June 2008), Opera (version 9.5, June 2008), Google Chrome (September 2008) and Safari (version 3.2, November 2008).

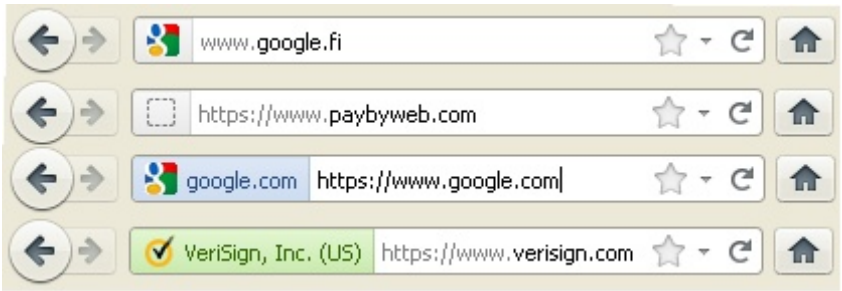

Figure 13. Security indicators in address bar of Mozilla Firefox 8 (from top to bottom: unsecured HTTP, ignored certificate error, regular certificate, extended validation certificate) [Article VII]

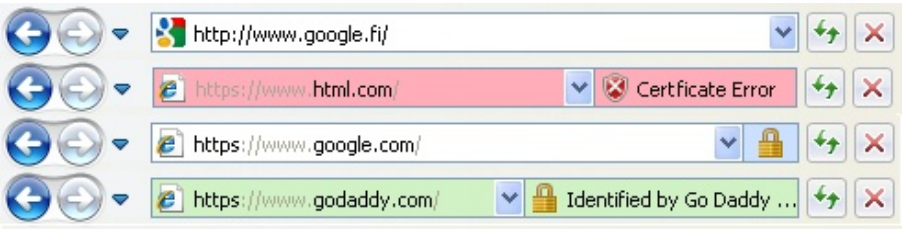

Figure 14. Security indicators in address bar of Internet Explorer 8 [Article VII]

The question whether the extended validation increase the security and trustworthiness has been considered by few researchers. Sobey et al. [73] studied whether users notice the additional trust indicators by tracking eye movements of 28 untrained test participants who were making online shopping decisions. They concluded that the validation indicators in Mozilla Firefox 3's address bar went unnoticed for all participants and proposed, as an alternative, more visible and obtrusive trust indicators. Similar results were gained by Jackson et al. [74] studied whether extended validation would help users to detect phishing attacks more easily with a test group of 27 participants and whether security trained users, who had read a help file, are capable to use these indicators. They noted that the trained users did not outperform the untrained users as extended validation did not help users to detect control attacks.

3.2.1.2 Limiting Certificate Issuers' Authority

Some researchers have addressed the problems of weak certification by proposing means to determine certificates' trustworthiness and to limit certificate issuers' authorities. Marlinspike presented [75] a solution called Converge for turning off all untrusted CAs in a browser. The idea includes a trust management scheme, where other users' views and consensus on particular CAs can be queried from notaries. Another solution called CertLock, presented by Soghoian and Stamm [76], tries to detect suspicious CA changes in certificates. They focus particularly on CA's country of origin and in the prevention of governmental attacks. CertLock uses browsers history information on certificates and warns end-users if CA's country of origin has been changed. In Perspectives [77], presented by Wendlandt et al., a trusted party collects issuer identity information frequently from TLS servers. The browser plugin may then query whether the issuer has been changed and warn end-user accordingly. A related certificate transparency proposal was made by Laurie and Langley [78]. They proposed

that end-users would accept only those certificates, which are available from trusted and public source. The approach would prevent long-life attacks, as service providers could to monitor this public source and suppress fake certificates, claiming their domain names.

3.3 Web Reputation

SSL certification provides mechanisms for checking that web servers belong to the legitimate entities. However, it does not address whether the server acts in appropriate and expected manner and thus whether the site can be trusted. Trust in WWW is based on users' perception on the trustworthiness of web sites as well as on reputation of services and service providers. To ease users to decide whether to trust a site or not, reputation services have emerged. These services enable clients to show visual warnings or block communication when connected to parties having a poor reputation.

The reputation is a measure determined by monitoring the behaviour and content of servers. Reputation provides a universal metric that can be used to assess trustworthiness of heterogenous and variable web servers. Reputation can be based on automated analysis or on ratings shared by users. Examples of systems where servers are evaluated using automated means include Google Safe Browsing [79], McAfee's SiteAdvisor [80] and Norton's Safe Web [81]. End-user based rating systems include peer-to-peer incentive systems (e.g. [82, 83, 84, 85, 86]) and web server rating services (such as PhishTank [87] and Web of Trust (WOT) [88]).

Untrustworthy web sites can be avoided by using blacklists, containing sites with bad reputation, and whitelists, containing sites with good reputation. Black- and whitelisting can be based either on automated techniques, where server's content is checked against malware fingerprints, or manual techniques, where users evaluate sites' trustworthiness. Human-based evaluation can be extensive only when a large number of people, a community or a crowd, are participating.

One of the crowd based reputation information providers is WOT. It is a company, which collects information from the open community of volunteers. These volunteers evaluate the web sites they visit by using browser add-ons, which are available for

Firefox, IE, Chrome, Safari, and Opera. The WOT company was founded July 2006. In November 2011 they reported that their database contains ratings from over 33 million servers.

The strength of WOT is in the detail of information. Evaluation is based on collecting users' subjective ratings, which vary from very poor (numeric values 0-19), poor (20-39), unsatisfactory (40-59) and good (60-79) to excellent (80-100). Ratings are given to four different categories:

1. Trustworthiness – whether the site is safe to use and free of malware and phishing attacks
2. Vendor dependability – whether the commercial actor (e.g., a web shop) behind the server can be trusted and provides good shopping experience
3. Privacy – whether the server is trusted to protect users' information appropriately and does not collect private information for vague purposes
4. Child safety – whether the server contains material such as adult content, violence or hateful language, not suitable for the children

In addition to the ratings, WOT provides confidence information for each rating. Confidence is presented by using six different categories and numeric value from 0 to 100. A rating is more credible when large amount of contributors have given similar ratings and when these contributors themselves have high individual confidence rating. Individual confidence ratings grow among time when users contribute. WOT does not reveal how the confidence ratings and reputation ratings are exactly calculated to make misuse harder.

Reputation systems are vulnerable for manipulating attacks as discussed by Moore et al. [12] who analyzed a phishing focused service called PhishTank [87]. They noted that the service is dominated by most active users and there is a risk of manipulation by small number of people. The accuracy, completeness and vulnerabilities of the WOT metrics have been analysed by Chia et al. [13]. They found that WOT was more comprehensive than the compared automated services (Google's Safe Browsing, McAfee's SiteAdvisor and Norton's Safe Web) in detecting malicious domains. They also argued that WOT may be resistant against manipulation attacks due to advanced

statistical analysis on the contributors' behaviour but that it is still vulnerable for determined malicious gamers. However, as manipulation is likely to affect only restricted amount of servers, it is not likely to distort large scale statistical studies.

Accuracy of crowd-based reputation systems and black lists has been enhanced by combining results from various heterogeneous sources. For instance, WOT utilizes blacklisting information from PhishTank. Use of quantitative web traffic information was proposed by Sharifi et al. [89], who automated information collection from various web services, including traffic ranking and search engine hits, and analysed how well this information supports scam detection.

3.4 Correlation between Certification and Reputation

Servers' support for SSL correlates with servers' security related reputation. SSL makes phishing and other masquerading attacks as well as confidentiality breaches harder. Therefore, it should increase reputation of servers when considering trustworthiness and privacy. The correlation and the causal relation between reputation and SSL are not straightforward or direct. In addition to SSL, other factors affect to the users perception of trust. A service provider that invests to security may also invest to other factors increasing the reputation. Nevertheless, the correlation can be used as one metric when evaluating the usefulness of SSL certification.

This subsection provides a large-scale empirical analysis on the correlation of SSL certification and community-based reputation evaluations. By using publicly available global certificate and reputation databases, the section studies how availability of SSL support and properties of certificates correlate to users' perception of trust, dependability, and privacy. The section proposes a metric for revealing the benefits that service providers gain from SSL certification in general, from authority selection, and from extended validation. The proposed reputation metric could provide a mean to quantify the users' valuation of security measures. Hence, it can be utilized when selecting and designing new web security mechanisms.

Existing work studying effectiveness of SSL certification and warnings in browsers has concentrated on experiments with restricted amount of participants. In this study, real

world data is analysed in much larger scale. In our study, the data comes from real deployments and thus cannot be distorted due to laboratory arrangements. The study has two implications. Firstly, we introduce a metric that can be used when analysing impacts and visibility of web security solutions. Secondly, correlation information is used to get some indications of the benefits that web services gain from SSL certification, extended validation, and selection of more reputable certification authorities.

3.4.1 Combining SSL Certificate, Web Reputation and Web Rank Data

We collected, combined, and analyzed data from three different repositories as illustrated in Figure 15. First we received SSL certificate database collected in SSL observatory project of Electronic Frontier Foundation (EFF). Secondly, information on web server's popularity was received in form of a list of top million servers produced by Alexa. Then, for the these valid certificates and for these top servers, we requested Web reputation ratings from WOT.

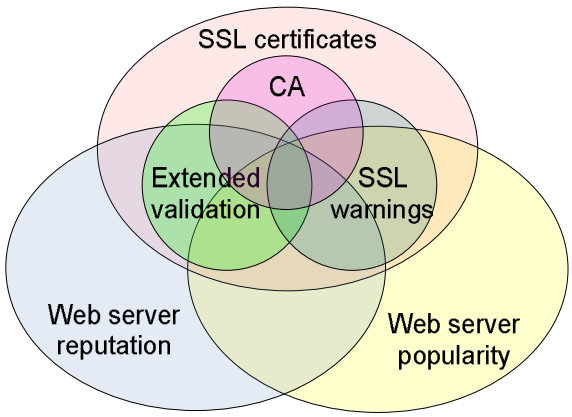

Figure 15. Composition of analysis data [Article VII]

SSL certificates available in the public Internet have been collected in EFF's SSL observatory project [67, 90]. The database contains almost 4 million certificates, including both 'regular' certificates as well as extended validation certificates. These certificates are certified by different certification authorities. We used those certificates, which were collected in December 2010 and classified as valid by EFF. For the analysis we resolved and selected those HTTPS servers, which had complete domain name (certificates with wild cards in domain names were ignored), were active and fully

working in November 2011. Services were classified as active if the request (to the root directory of the SSL (443) port) resulted a reply larger than 1kB. This limit filtered most servers were HTTPS port is used only for redirection to HTTP port or for some other limited purpose.

Information on the most popular web servers were received from Alexa, which is a web service providing a list of top million web services [91]. The list was used to get domain names of servers, which are really used and frequently visited. This enables comparison between HTTP only servers and servers with HTTPS support. For each server in the list, we collected HTTPS status information indicating whether the HTTPS port was open and whether the connection succeeded without warnings.

WOT reputation metrics were collected for all HTTPS sites as well as for HTTP only sites among top million servers in order to enable comparisons. In our analysis, described later, we used only those ratings with reasonable confidence value (12 or higher).The confidence limit does not affect substantially to counted averages but it filters out some suspicious ratings. Data was collected and analyzed with Linux shell and Perl scripts. SSL status queries and certificate verifications were done on a client based on OpenSSL. Certificates of contacted servers were verified against root certificate list used by Mozilla. MySQL was used as database software. For EFF dataset we found 201,099 active and reputed HTTPS servers and for Alexa dataset we found reputation information for 132,533 HTTP only servers, for 68,961 HTTPS servers, and for 34,985 broken HTTPS servers (showing security warnings when connected).

3.4.2 Correlation Results

3.4.2.1 Does HTTPS Support Increase Reputation?

The effect of HTTPS support to reputation rankings was studied by calculating average and distribution of reputation values from the Alexa dataset, which contained information from top million servers. The results for trustworthiness and privacy reputation are given in Table 3 and Table 4, respectively. For both metrics the rating for errorless HTTPS support gives around six additional per cents. Similarly, the amount of poor and very poor rates drops from around 9% to 4% when HTTPS was supported. Additionally we studied how the security warnings, such as domain mismatch or self-

signed certificate, affects the ratings. We noted that HTTPS increases trustworthiness only when used correctly. However, even misused SSL based cryptography increases privacy ratings with one point.

TABLE 3. TRUSTWORTHINESS REPUTATION OF SERVERS WITH AND WITHOUT SSL SUPPORT AND WITH BROKEN SSL SUPPORT SHOWING WARNINGS [ARTICLE VII]

Server type / count	**Average**	**Distribution (%)**				
		Excellent	*Good*	*Unsatisf.*	*Poor*	*Very Poor*
HTTPS / 13,497	84,7	84,5	9,5	1,8	1,0	3,1
Broken HTTPS / 9,483	78,7	73,1	13,4	4,1	2,5	7,0
HTTP only / 41,250	78,6	72,1	13,8	5,0	2,5	6,5

TABLE 4. PRIVACY REPUTATION OF SERVERS WITH AND WITHOUT SSL SUPPORT AND WITH BROKEN SSL SUPPORT SHOWING WARNINGS [ARTICLE VII]

Server type / count	**Average**	**Distribution (%)**				
		Excellent	*Good*	*Unsatisf.*	*Poor*	*Very Poor*
HTTPS / 13,001	84,9	86,0	8,1	2,0	1,1	2,8
Broken HTTPS / 8,776	80,0	73,7	13,1	4,9	2,4	5,8
HTTP only / 37,197	78,9	73,4	13,0	6,6	2,8	6,2

The servers in HTTPS category may have also the HTTP port open. Hence, we cannot say whether the user evaluations were done in the HTTPS secured connection or not. From the larger EFF dataset, we found servers that had only HTTPS port active. For 431 servers the average trust value was 86,6 (when the average value for all HTTPS servers in 'EFF dataset' was 85,8). The privacy ratings for 371 servers were 87,9 (and 87,1 for all). This small sample indicates that reputation of servers supporting only HTTPS would be even larger.

We studied also how trustworthiness and privacy reputations correlate with the popularity of server. Sliding averages presented in Figure 16 illustrate that the better ranking in Alexa increases trustworthiness and privacy value. The difference of reputation between secured and unsecured is visible despite the popularity, though the difference is smaller with more popular servers.

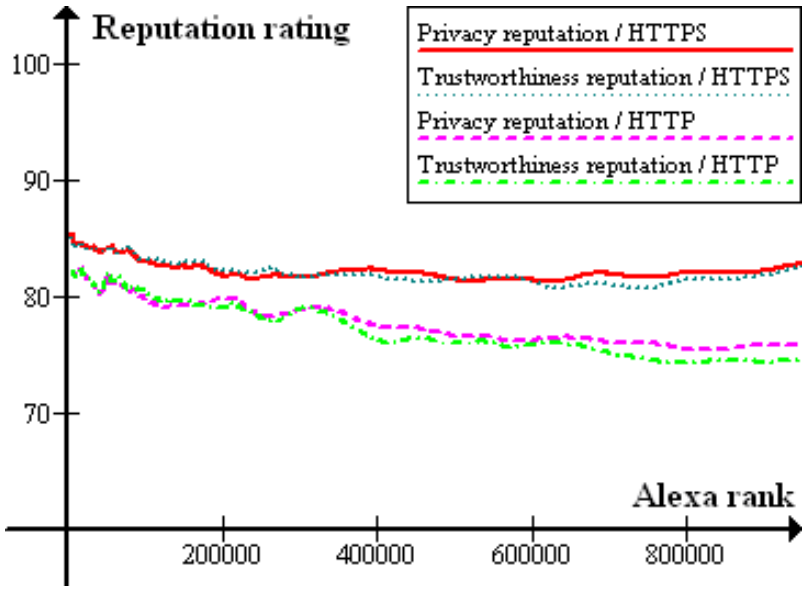

Figure 16. Dependency between reputations and popularity [Article VII]

3.4.2.2 Differences between CAs

There are clear differences between the reputation of servers certified by different CAs. Table 5 presents results of CAs, which all had more than thousand valid certificates used by active and trustworthiness ranking with reliability at least 12 points servers within 'EFF dataset'. The results show a difference of over 10 points between the averages of the best and the worse CAs. The differences between CAs are illustrated in Figure 17.

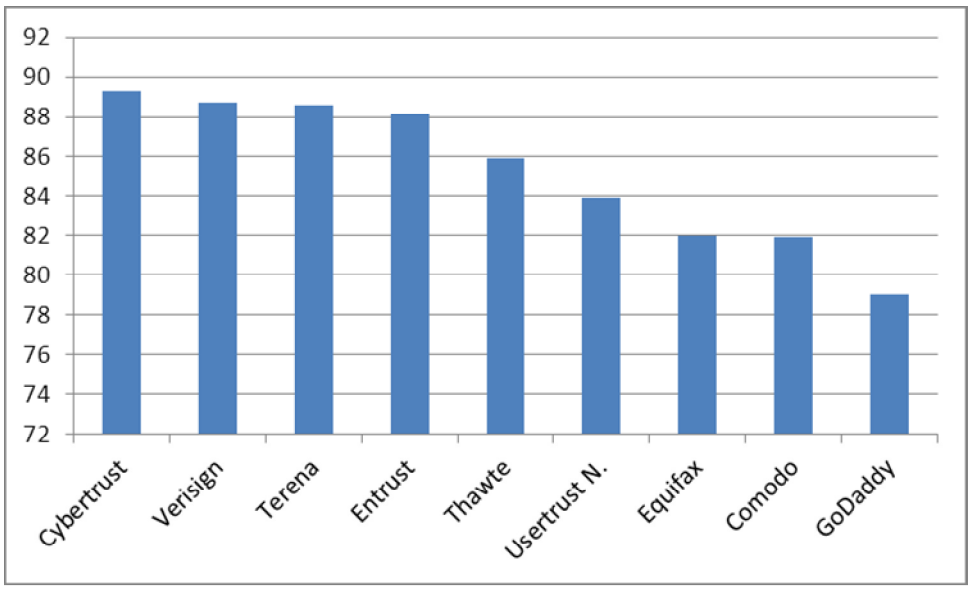

Figure 17. Trustworthiness ratings among servers with certificates from different providers

The difference is even significant when looking at the ratio of poor and very poor sites: increase from close zero to 7,4%. Different CA brands provided by one company have not been combined in the table. E.g., Comodo is also the provider of The Usertrust Network and Terena certificates, Symantec is the owner of Verisign and Thawte.

TABLE 5. TRUSTWORTHINESS REPUTATION OF SERVERS CERTIFIED BY DIFFERENT CAS [ARTICLE VII]

CA / certificate count	**Average**	**Distribution (%)**				
		Excellent	*Good*	*Unsatisf.*	*Poor*	*Very Poor*
Cybertrust / 1061	89,3	96,6	3,0	0,2	0,0	0,2
Verisign / 9993	88,7	92,1	6,0	0,8	0,4	0,7
Terena / 1410	88,6	95,7	4,3	0,0	0,0	0,0
Entrust / 1747	88,1	92,8	4,6	1,4	1,0	0,2
Thawte / 5506	85,9	85,3	10,7	1,6	1,0	1,3
Usertrust N. / 1994	83,9	77,4	18,7	1,0	1,0	2,0
Equifax / 4828	82,0	74,0	19,0	1,9	1,3	3,8
Comodo / 1557	81,9	75,8	16,2	2,1	0,7	5,3
GoDaddy / 2973	79,0	67,5	22,7	2,5	1,8	5,6
Total / 39482	*85,8*	*84,6*	*11,6*	*1,2*	*0,8*	*1,8*

3.4.2.3 The Value of Extended Validation Certificates

Extended validation provides only small or no increase of reputation at all. Table 6 compares average trustworthiness and privacy values of EV certificates to non-EV certificates within the EFF dataset. Diagrams in Figure 18 illustrate how the ratings are distributed. Trustworthiness average is 0,7% higher and privacy value is 0,5% smaller.

TABLE 6. TRUSTWORTHINESS AND PRIVACY REPUTATION OF SERVERS WITH REGULAR OR EXTENDED VALIDATION CERTIFICATES [ARTICLE VII]

CA / certificate count	Count	Average
Trustworthiness		
Regular	36297	85,7
EV	3185	86,4
Privacy		
Regular	32166	87,1
EV	2839	86,6

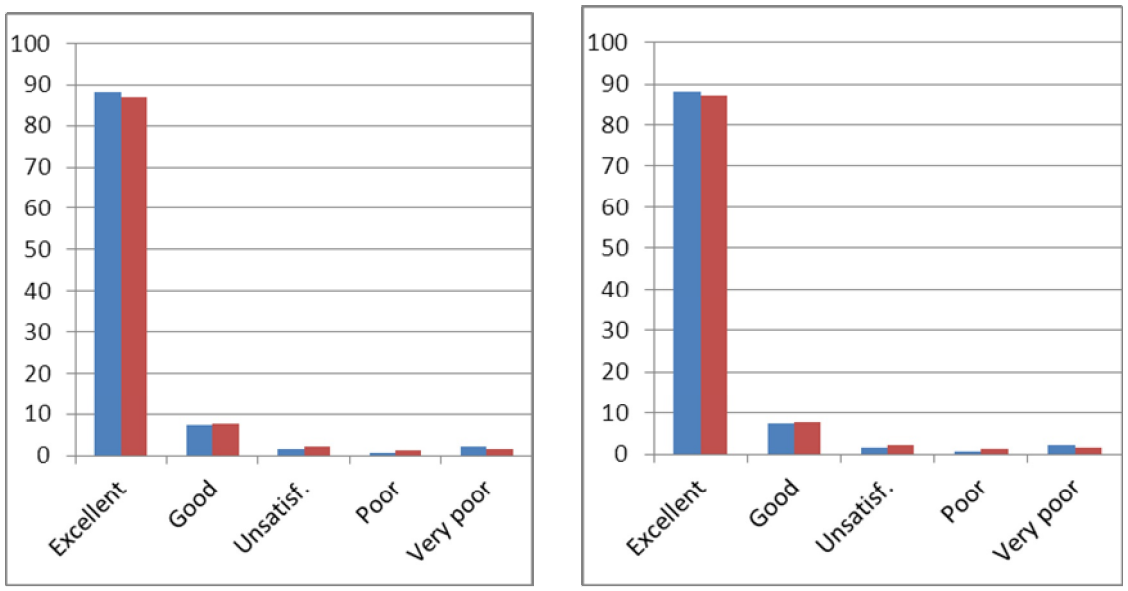

Figure 18. Distributions of trustworthiness ratings (left diagram) and privacy ratings (right diagram) the in each category the left (blue) bars indicates servers with regular certificates and the right (red) bars servers with EV certificates

Table 7 describes CA specific trustworthiness ratings for CAs with more than 100 EV certificates. When comparing to CA specific numbers to generic CA results in Table 5, there is a small increase of reputation all CAs except for the largest EV provider. For

instance, For Verisign the EV rate is 0,7% smaller than the rate for all Verisign certificates.

TABLE 7. TRUSTWORTHINESS REPUTATION OF EV CERTIFIED SERVERS BY PARTICULAR CAS [ARTICLE VII]

CA / certificate count	**Average**	**Distribution (%)**				
		Excellent	*Good*	*Unsatisf.*	*Poor*	*Very Poor*
Cybertrust / 255	89,9	100	0	0	0	0
Verisign/ 1688	88,0	91,0	5,3	1,9	3,5	0,9
Thawte/ 183	86,2	85,2	8,7	3,3	33,9	0,0
Comodo / 226	83,2	81,0	11,5	0,9	6,6	4,9
Globalsign/ 366	83,1	70,2	25,7	1,9	2,2	1,1

The results presented in this Section are discussed in Section 7.1. However, it is worth to emphasize that correlation does not imply causal relationship.Trustworthy sites typically utilize stonger security mechanisms and use of strong security mechanism may make site to be more trusted. However, these statistics shows a snapshot of the current status in other words the results show what is the correlation between reputation and use of SSL and certificates in autumn 2011.

4 Platforms and Ecosystem for Secure Interoperable Home Environments

This section studies what authentication and authorization mechanisms are needed and available in existing network platforms and frameworks, which facilitate the interoperability and ease service development. The section focuses on security requirements from the point of view of home networks with various cooperating devices and software components. The section also studies the ecosystems for home security by presenting roles that different parties from developers to standardizers and from operators to end-users may have. The requirement survey extends a study presented in Article II. The section contributes by proposing taxonomy for authorization solutions and by describing experiences with a secure middleware platform implementation, called OpenHouse. The OpenHouse approach, initially presented in Article III, illustrates the need for adapters to enable interoperability and also promotes authorization model based on third-party certification for achieving easy-to-use but fine-grained security control.

4.1 Security Needs in Home Networks

4.1.1 Networked Homes

Home networks have in the recent years become common. The first motivations for home networks were the sharing of printers, data storage, and broadband internet connectivity between different computers of home residents. Recently, more versatile networked devices and services have emerged. For instance, we have seen networked cameras, video recorders, high definition televisions, mobile phones, game consoles, sensors, robots, exercise devices, toys, climate control equipment, and energy production systems. Local services, provided by these devices, and remote services, provided by different services providers, organizations, and enterprises, have become available for different kinds of user terminals in homes. Some examples of networked devices for homes are listed in Figure 19. In the future, more and more advanced services and applications are expected to emerge. These devices are becoming more autonomous and they will cooperate over networks without any user interactions.

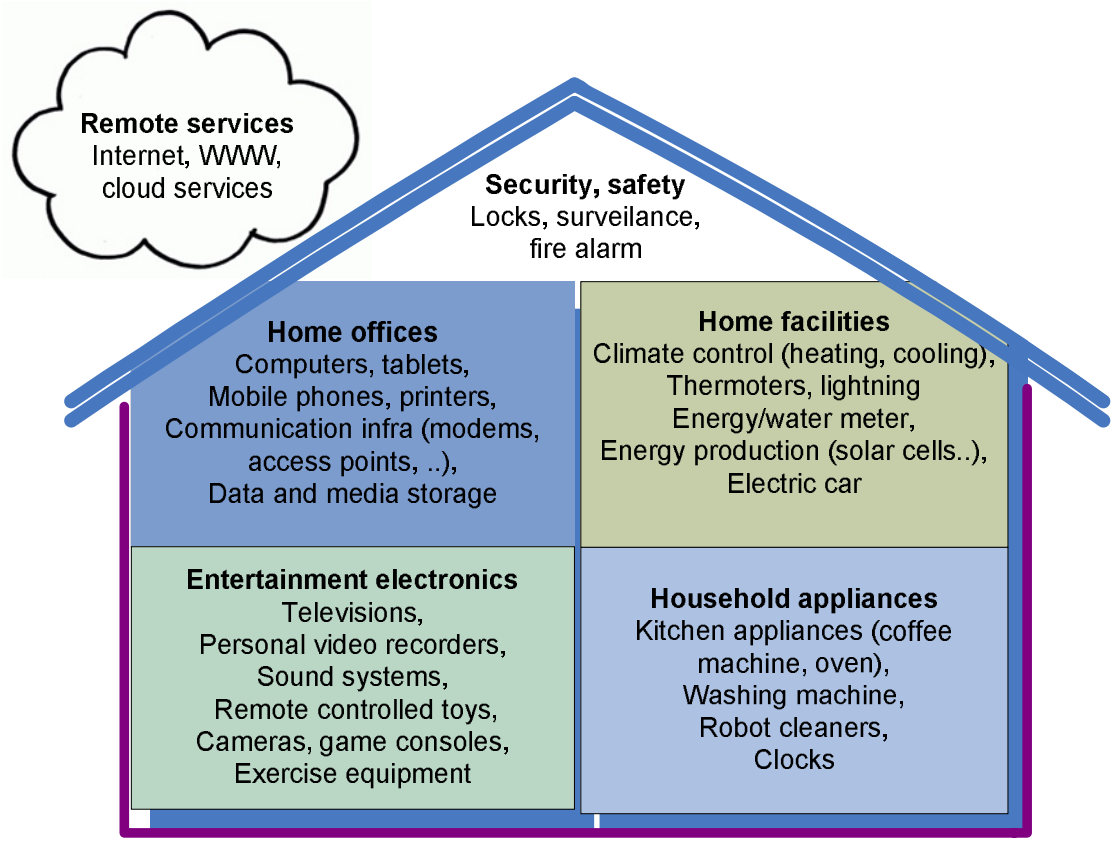

Figure 19. Networked homes consist of various heterogeneous devices

The trend of networking homes has been enabled by the development of different networking technologies and service gateways, enabling communication between heterogeneous devices and device types. The use of open networking standards as well as open hardware and software based service platforms have made introduction of new home services and devices easy for manufacturers and software developers. Consequently, new kinds of applications and services using large amount of cooperative devices can be introduced rapidly for the consumers.

4.1.2 Motivations for Authentication and Authorization

The new technological advantages in homes have also introduced new challenges, which must be addressed before the potential of networked home devices can be achieved. Particularly, questions related to security and privacy and also to reliability, safety, and usability remain partly unsolved. Security issues in home networks are emphasized as homes contain large amount of safety and privacy critical assets.

In open multi-user home network environments, we have several motivations for introducing fine-grained authentication and authorization mechanisms. One motivation is ability to withstand *malicious attacks*. Home networks consist of programs and devices, which come from different sources. Every program and device cannot be completely trusted to behave correctly. With fine-grained control, it is possible to neutralize threats, which malicious programs pose towards services inside homes. Different components and communicating technologies provide different security levels. Consequently, some components are less able to withstand security attacks. Remote attackers may easily gain an access to home networks, since many programs inside home communicate directly or through firewalls with counterparties locating in the Internet. Alternatively, attackers may be in the close proximity of home network and utilize e.g. weaknesses of wireless devices. Attackers with access to an already compromised device may try to gain control over other home devices. Hence, multi-level protection is needed to prevent single security breach to jeopardize whole home network.

The management and upkeep of home network infrastructure has attracted interest from the research community. In [92] Grinter et al. studied the effort required to setup and maintain such networks over a longer period of time and found that they are surprisingly complex and their upkeep can involve contacting multiple external parties such as ISPs and cable operators. *Unintentional modifications* of the settings of a home router for instance, can easily lead to the whole network becoming unusable for a long time. Rodden and Benford [93] point to the fact that the burden of creating a networked home often falls on the shoulders of non-technical inhabitants. This can be a very time consuming process. In a paper which outlined security requirements for a widely used home networking standard, Universal Plug and Play (UPnP), Ellison [94] motivates the need for different granularities of security in a home in order to prevent such unintended modifications and emphasizes that the *social structure of the household will have implications for how access to resources on the home network needs to be controllable*. In many situations, social control and good manners are not enough to restrict that every device is used in appropriate manner. Homes with children, siblings and quests are clear examples where usable fine-grained authorizations are needed.

Protection is needed for various types of interactions. In some cases it is necessary to protect interfaces or particular pieces of information. Information on the availability of services is also important in home environment. Services should be visible only for those services and users who are able to use them. This measure prevents reconnaissance, protects privacy, and may improve usability as inaccessible services are hidden from lists. Maintaining privacy of homes is important to secure our social relationships but also to keep homes uninviting for burglars.

Access control is usually thought of as a mechanism for keeping named, server-side, resources private and confidential from clients but it can also have other positive implications for users. Brush and Inkpen [95] examined the shared use of technology in households and identified that profiles on devices such as PCs are often used to distinguish between family members. In their study of 15 U.S. households, they found that such access control was used to *personalize the user experience* of the PC rather than to keep content private within the family. So from the point of view of a networked device manufacturer, supporting some level of access control can have two major benefits. In addition to being able to prevent damage to the device (by limiting who can change critical settings), knowing who it is that wishes to use the device can be an important way to personalize the user experience – for instance a video recorder, supporting user profiles, would know which programs to recommend based on the taste of the current user.

4.2 Authentication and Authorization in Network Middleware for Homes

In home networks, middleware solutions have been seen as one approach to solve interoperability, connectivity and security issues caused by the complexity and heterogeneity. Middleware is a broad term, which can refer to common protocols on top of connectivity mechanisms (i.e. protocols in OSI layers four to six) as well as services facilitating the interoperability. Middleware solutions have been proposed e.g. to ease in service discovery as well as to make communicating programs independent of the platforms and communication protocols. Figure 20 illustrates a typical home architecture with services and client software components, different network interfaces,

common middleware solutions, an execution platform and a gateway enabling interoperability.

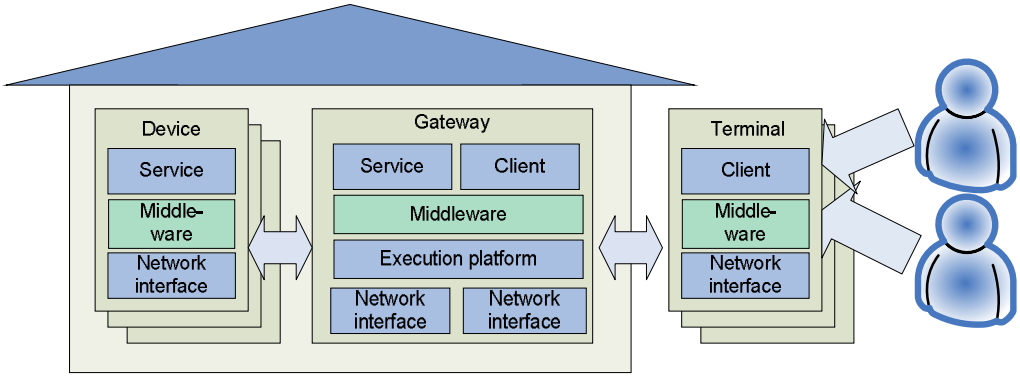

Figure 20. A gateway and middleware-based architecture for SOHO services [Article II]

Authentication and authorization solutions in home networks are typically based on the communication protocols with cryptographic and key establishment mechanisms such as the ones described in Section 2. However, these security mechanisms may not be enough to secure interactions based on middleware. Hence, different security mechanisms have been introduced also for middleware approaches. Alas, in the context of home networks, there is currently no single best approach for authentication and authorization. Also, there is no universal solution for handling complexity caused by heterogeneity of devices. In homes, security needs can be very fine-grained and there is a need to consider various parameters, which may depend on the contexts, environments and technologies in use. Due to usability and costs, the current solutions have typically confined themselves to limited use cases or to coarse-grained access control.

This Section 4.2 surveys how four prominent Small Office Home Office (SOHO) technologies, namely OSGi, Windows Networks, UPnP, and WPWS, fulfill authentication and authorization requirements. Further, the section underlines some potential gaps and needs for future solutions.

4.2.1 Classification of Authorization Solutions

Authorization can be based on different architectural solutions. The main design question is where are the authorization decisions made and where are the authorization policies store (or particularly how much authorization information do client devices store and how much do service devices store). **Figure 21** illustrates taxonomy, which can

be used to classify different authorization solutions. Taxonomy is divided into three main categories. In *trusted-authorizer based decision making* category, the authorization decision and policies are made by a trusted party or parties such as centralized access control components. The servers are only required to perform simple operations when enforcing access control. In *distributed authorization models*, the access control decision is the responsibility of individual devices providing services or gateway devices controlling access to these devices. In these models, centralized component does not have to be involved in any manner. The *hybrid-models* category requires some involvement from a trusted component as well as some non-trivial decision making from servers or gateways. Practical implementations of all of these categories may be linked to distributed or centralized means to configure and provide user input and centralized decision.

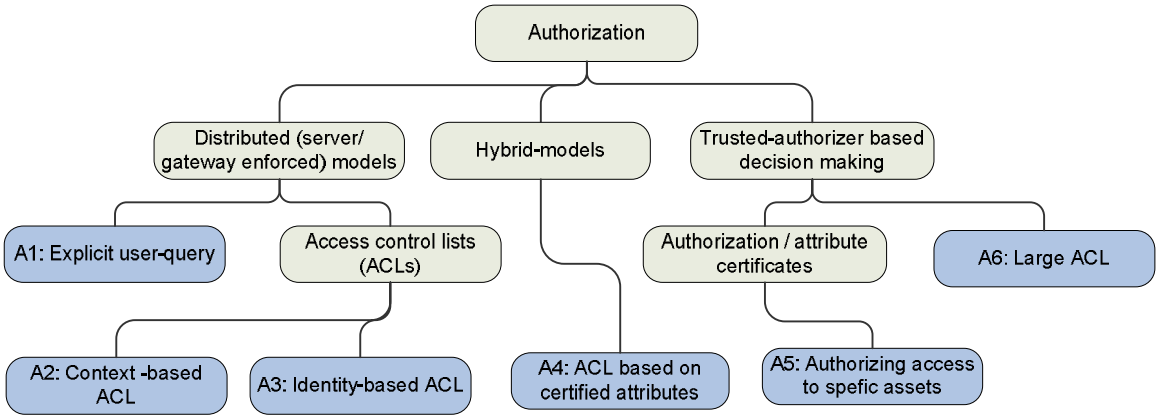

Figure 21. Taxonomy of Authorization Solutions for Network Architectures

Distributed models have been divided to two categories according to the requirement for end-users' involvement. The authorization may be *explicitly queried* (A1) from end-users with sufficient permissions, when a particular action is performed and permissions are needed. The authorization may also be configured beforehand using access control lists (ACLs). In the ACL based alternatives, client devices provide information proving their own identities. All policy information is kept on the server side, i.e. inside the devices being controlled. The context in *context-based ACL* (A2) can refer to any security relevant attribute relate to the client or operation. For instance, environmental context (time of date or location of client) or trust-related context (reputation of the client or software the client is using) can be used when making authorization decisions.

Identity is a special case of context. In *identity-based ACL* (A3) user's identity is tied to access permission.

Trusted-authorizer based models are divided into two categories. A trusted authorizer may maintain a *large ACL* (A6), which contains authorization information for each service and device. Devices may then query this authorizer when a particular action must be authorized. Trusted authorizer may also *authorize actions on specific assets* (A5) by providing authorization or attribute certificates to clients. These certificates are cryptographically protected tokens, which authorize clients to perform particular action on particular services. The trusted authorizer may be a centralized administrator or a peer device delegating it own permissions.

Hybrid models combine characteristics of distributed ACLs and authorization certificates. In *ACL based on certified attributes* (A4) model, a client is given a certificate proving that a client has a particular authorizing attribute. Distributed devices and services then enforce and check from ACL whether that attribute authorizes a particular action on a particular asset.

Table 8 compares the feasibility of the alternatives by listing pros and cons and by analyzing actions which occur when new users and terminals are coming to home or when security policies change. In the table alternatives are compared by characterizing how expensive typical operations are.

Table 8. PROS AND CONS WITH ALTERNATIVE AUTHORIZATION SOLUTIONS

A1: Explicit user-query
+ Deployment of new devices and services is easy as all authorization related decisions must be done when devices are used – This approach does not scale well as the number of devices and services increases in the home. The amount of user queries will start to affect the user experience. + Suitable for security sensitive operations occurring seldomly. – Is not suitable for autonomous solutions. A user with capabilities to authorize actions must be present when an action occurs.
A2 and A3: Distributed identity-based ACL
– Updating authorization policies is costly as information must be pushed to each relevant service (these services must be accessible during this operation) + Deploying new services is cheap (only relevant security policies and a list of authorized users must be pushed to a device, which is hosting the service)

A4: ACL based on certified attributes
+ Adding new users and assigning roles is easy (as only one client device must be delivered information) + Deploying new services is easy (only relevant security policies must be pushed to a device, which is hosting the service) – Revoking role assignments is difficult: Revocation lists must be pushed to every relevant device. Alternatively, authentication information may be valid only for short period of time (services must have up-to-date clock and there must be a server for updating role authentication certificates) – Changing policy information is costly (as each relevant device must be updated)
A5: Attribute certificates authorizing access to specific asset
+ Adding new users and giving permissions is easy (as only one client device must be delivered information) – Deploying new services is costly (information must be pushed to each relevant client device) – Revoking permissions is difficult (as revocation lists must be delivered to each relevant device or as there must be available certificate server). Consequently, life time of certificates is typically limited, which causes requirements (up-to-date clocks and services for certificate renewal).
A6: Large centralized ACL
+ Adding new users and services is straightforward as only one device must be connected + Giving and revoking permissions is easy as only one device must be connected – Centralized element typically requires a significant investment – The required central element makes home network dependent on centralized elements, which must have sufficient resources, which must be available and dependable, and which, for instance, must have powered all the time

4.2.2 Existing Frameworks and Middleware

This section describes authorization solutions in few frameworks for applications and devices in home networks. The list is not comprehensive. Instead, the purpose is to illustrate the current status in the prominent commercial-off-the-self products.

4.2.2.1 OSGi Security

Open Service Gateway initiative (OSGi) [96] is a platform for executing and deploying Java services and interoperability gateways. It provides solutions both for software authorization as well as for user and remote device authorization. Software authorization features of the OSGi are based on the Java security model. Particularly, OSGi enables authentication of downloaded software components by checking package signatures when installing programs. Further, OSGi specifies Framework Security

Manager, which can be used to enforce that programs, performing critical actions, have required permissions.

For authentication and authorization of users and remote devices, OSGi specifies User Admin Service. This service stores credential information enabling authenticators to authenticate users and devices. Furthermore, the service provides authorization objects, which are appended to service requests so that software bundles, providing services, can check if requests are authorized. The authorization model of User Admin Service is based on role-based access control (RBAC) [97]. Authenticators can be components delivered with a gateway implementation like a HTTP(S) server or custom components such as Session Initiation Protocol (SIP) service. For instance, in [98] the thesis author incoperation with coauthors described architecture and a prototype for controlling home lighting appliances remotely with Session Initiation Protocol (SIP) extension. The remote SIP communication can be secured with TLS.

Authorizations of service accesses may also require that the user is interactively queried for acceptance, that access control query is send to a remote device, or that particular contextual condition is met. To enable dynamic condition checks, OSGi provides Conditional Permission Admin service. This framework enables service developers to program custom security checks, which will be executed when service objects are accessed.

Security solutions provided by OSGi have configuration demands, which are often too laborious and difficult for common users. Also, even when security solutions are in use, there are many remaining risks including:

- An attacker with access to the underlying operating system or hardware can circumvent all security mechanisms.
- Malicious software may be installed and given large privileges e.g. because software verifying signatures does not give understandable warnings or because the user ignores risks.
- Complexity of configuration may yield security holes. For instance, if only users are authorized and not software, untrustworthy software may misuse users'

privileges. Also, OSGi authorizes only a client making request. If an attacker requests the client to access a service on its behalf, an intrusion may succeed.

- Critical services may not protect their assets carefully e.g. due to weak implementation or design.
- OSGi and Java security model are vulnerable to threats, which are related to availability of resources. Once access to a resource is granted, a program can use it extensively.

4.2.2.2 Kerberos

Kerberos [99] is a client-server based approach for mutual authentication as well as for authorization. Kerberos is based on symmetric key cryptography and requires a trusted server. The basic steps of the protocol are the following. First, a client authenticates to an authentication server once using a long-term shared secret (a password). The key establishment can be classified as authenticated symmetric crypto key agreement – see **Figure 4:P2**. A client sends a one-way hash from a password to the server. Then, as a reply the client receives a Ticket Granting Ticket from the server. Later, when the client wants to contact some service, it can (re)use this Ticket Granting Ticket to get service tickets (with short life time) from authorizing server (ticket granting server). The latter tickets can be used to prove authentication and authorization to the service. Kerberos supports different cryptographic protocols. The used algorithm is negotiated automatically between the client and servers.

4.2.2.3 Windows Network Security

Microsoft's Windows operating system provides authorization features, which are usable for controlling users and programs behavior inside personal computers. For networked homes, Microsoft has incorporated mechanisms for authenticating users and devices. Further, there are proposals for extending control of programs behavior to networked systems.

Authentication and authorization in Windows Networks is based on Active Directory [100], which is a centralized configuration, authentication, and authorization service for Windows networks. The active directory (AD) is based on Kerberos, which was presented Subsection 4.2.2.2. AD is used for authenticating the end-users and devices

and to control access to resources and files. Windows operating system provides also an access control solution [101], which authorizes programs access to files and resources with the computer. The access control mechanism enables fine granularity access control over different types of operating system components. The same access control mechanism is used by all system components including the file system, kernel objects as well as user interface objects. Every object requiring protection is assigned a security descriptor, which stores owner, group, ACL, and auditing information. ACLs are containers for access control entries (ACEs). ACEs determine which access rights are granted for particular users. ACEs contain 16 bit long access mask specifying the access rights, such as list directory, add file, and read attributes for directories. However, this control is only within those devices that are hosting the programs. Windows designers have also proposed [101] a mechanism for extending programs' authorizations to remote devices. The mechanism utilizes Kerberos protocols field, authorization-data, to limit clients' authority in the remote Windows devices. When a process with restricted context authenticates to a remote device, the Kerberos stores programs restrictions, i.e. restricted context, to a Kerberos ticket. The remote party then extracts this information before the remote server process is allowed to act on behalf of the user.

The main disadvantage of Centralized authentication and authorization solutions such as Kerberos is that the security server must be dependable and always available. The advantages of Kerberos include reliance only on symmetric cryptography making it computationally less expensive than solutions relying on asymmetric authentication mechanisms.

In addition to Kerberos based access control, Windows 7 introduced more lightweight HomeGroup [102, 103] concept for easily configuring permissions for different Windows devices within home networks and for sharing services with devices outside the Kerberos domain. Home group is a virtual private network where users within particular group can access devices, files and services shared in that group. HomeGroup devices authenticate using Microsoft's Public Key Cryptography User-to-User (PKU2U) protocol [104]. The access to home group is based on group specific random password generated by Windows.

4.2.2.4 UPnP Security

UPnP v1.0 is a network architecture and interface specification enabling interoperability between various UPnP compatible devices. It provides security mechanisms [94, 105] for protecting communication between UPnP devices as well as for authenticating and authorizing service accesses.

The specification secures control messages by proposing use of XML Signatures to achieve integrity; symmetric encryption algorithm (AES) to protect confidentiality; and sequence numbers to prevent replay attacks. Authentication between devices is based on security ID, which is a (SHA-1) cryptographic hash from device's public (RSA) key. Proposed association model for adding new devices to the network requires the user to manually ensure that the ID, which was delivered with the new device e.g. in a printed form, is made correctly available for the network.

There are two alternatives proposed to enable authorization: access control lists (ACLs), which locate in devices; and authorization certificates, which clients (UPnP control points using services) must acquire. Authorization for control points to access services is given by a security console, which edits ACLs or grants authorization certificates. Each device has also a secret password, which must be known to a security console before it can take the ownership of a device and modify device’s ACL. This password should be device specific, should be able to withstand guessing attacks and may be e.g. on a label in a device or displayed by a device.

4.2.2.5 DPWS security

Device Profile for Web Services (DPWS) [106] is a Web Services specifications based alternative or replacement for UPnP. DPWS specification proposes that X.509 certificates and TLS protocol are used to authenticate and secure communication between DPWS devices. To secure authenticity of service discovery XML Integrity signatures as specified in OASIS Web Services Security (WSS) specification can be used. This limits attackers' potential to perform DoS attacks, as unsigned messages are not processed. Also, authentication of services, before communication sessions are created, minimize the threat of bogus services. How servers are given credentials (e.g. certifications from a trusted party), proving that they are permitted to advertise

particular services, is not specified. Confidentiality of discovery messages is not addressed.

The issue of authorization, i.e. controlling what authenticated devices are able to do, or associating new devices with a network are not in the scope of the specification. Authorization solutions available for TLS are potential also for DPWS. For instance, TLS may utilize attribute certificates, specified in X.509 Internet Attribute Certificate Profile for Authorization [107].

4.2.3 Authorization Requirements for Home Middleware

This subsection identifies access control needs not answered by the existing middleware approaches, which were presented in the previous subsections. Essentially, this subsection presents requirements and research approaches for making authorization mechanisms in home environments more easy-to-use and autonomous.

4.2.3.1 Management of Heterogeneity and Security Levels

Many home networks consist of several networking technologies and security solutions. These technologies have different kinds of security properties and hence, variable security levels, which means strength and resistance against different security threats. There is a need to enable use of different technologies but at the same time control that assets are not compromised due to simultaneous use of weaker devices and protocols.

The management over heterogeneity requires that there are means to compare and valuate the security strength of different mechanisms. Consequently, there is a need for systematized means to quantify the security levels. For example, Table 2 in Section 2 presents some metrics which can be used to compare security strength of key establishment protocols. Those measurements describe protocols strength against active and passive exhaustive search attacks. Other metrics are needed to measure and compare other security relevant characteristics. Surveys and taxonomies related to security metrics include e.g. [108, 109].

The security level information is utilized in mechanisms and architectures, which control how different devices may cooperate. Several security middleware solutions, which monitor and consider peers' security capabilities and requirements and are able to

dynamically adapt their behavior accordingly, have been proposed. These middleware solutions either utilize and complement or replace transportation layer specific security protocols. Zhuge et al. [110] studied what security mechanisms are needed and available for wireless home networks. They proposed centralized (Kerberos-based) architecture, which addressed devices' heterogeneity and different security needs by supporting different security levels. They also proposed that low capacity devices could delegate security functions to other devices. In addition to solutions where authorization decision is made by trusted authority, security enforcement based on security levels can be easily used to distributed ACL based models including 'publish and subscribe' architectures. For example, the Genetic Messaging Oriented Middleware (GEMOM) project has proposed [111] middleware and mechanisms for adapting security according to peers' requirements. Secure Middleware for Embedded Peer-to-Peer Systems (SMEPP) [112] has focused on the secure cooperation between embedded devices. SMEPP is able to adapt security levels according to devices' capabilities and needs. In Subsection 5.2.4, we present a security-level based authorization solution, which supports semantic web technologies.

4.2.3.2 Intuitive Configuration of Policies

It would of course be possible to design very fine-grained control of which service actions each device and user would be allowed to make, for instance which family members would be allowed to tune a television. However, due to the large amount of services and users, this leads to complexity, which is difficult for ordinary non-expert users to manage. Therefore, in a system with a large amount of access control subjects and objects, fine-grained policy configuration will be a challenge, which affects usability. Simple solutions where users are required to configure user names and passwords do not scale well as the number of devices increases in houses.

One possibility to ease configuration work is to classify users, devices, and programs into groups, which give them different permissions e.g. to advertise or access services. Also, services or devices can be classified to groups, which require particular permissions before they can be accessed. For instance, in Linux systems a file may be executable for a particular user or for every user belonging to the same group as the file. A prominent more general and flexible classification scheme is the ***role-based access***

control (RBAC) [97, 113]. In RBAC, grouping is done by giving access control subjects into roles, which can be defined so that they are meaningful and intuitive for typical users. Roles may form hierarchies to simplify configuration. There may be roles for users and for programs. The user roles are targeted for restricting users' access to particular services; whereas, program roles can be used for protecting the integrity of system software and for sandboxing untrustworthy programs.

Similarly to access control subjects, also access control objects (i.e. the accessed resources and assets) can be classified. Examples of operating system level models where device's resources or interfaces are grouped into a handful of static categories to which permissions are tied include Posix capabilities for Linux systems [114] and capabilities in Symbian operating system [115]. ***Domain and type enforcement*** (DTE) [116, 117] is an access control model where subjects (e.g. processes executing programs) can be more flexibly grouped into domains and objects (e.g. files) types. Network-level DTE [118] extends this software authorization paradigm from operating systems environment into networks. DTE treats network packets as objects. Only processes belonging to particular domain can send and receive packets. Each packet carries a label, providing information of sender's domain and packet's type. Unlabeled packets coming from nodes, which are not DTE compatible, must be labeled in the receiving end e.g. according to sender's address.

Authorization policies may be challenging to configure before hand and they may not cover all potential access situations. Therefore, run-time policy configuration mechanisms are typically needed to handle cases where an unauthorized client accesses a protected service for the first time. Ka-Ping Yee [119] instructed that authorization should be implicitly derived from end-users actions. ***Implicit authorization*** means that the program gains access permission to particular asset only when the user explicitly uses that program to access the asset.

The behaviour of security solutions can be controlled based on the context i.e. on temporal situation or environment. Existing research efforts on context-aware security include ***context-dependent access control models***. Covington et al. [8] extended the role-based access control model by representing contexts with a new type of role called

environment role. Environment roles capture relevant environmental conditions that are used for restricting user privileges. Permissions are assigned to roles (both traditional and environmental ones) and role activation/deactivation mechanisms regulate the access to resources. Toninelli et al. [10] presented a context-aware policy model where context is any characterizing information about controlled system entities and about their surrounding world relevant for enabling entities to operate on resources. Intuitive location-inspired access control models include a concept of virtual walls, proposed by Kapadia et al. [120]. Virtual walls enable users in pervasive environments to protect and control their digital privacy by protecting their virtual assets using concepts familiar from the physical world.

4.2.3.3 Trust Management based on Past Behavior and Contribution Tracking

Trust management solutions provide potential mechanisms for a system to learn authorization policies without requiring them to be explicitly configured. In trust management solutions, peers previous behavior is tracked or monitored and based on the collected trust information devices are able to autonomously decide whether cooperation with the peer or server should be allowed. Similar concepts have been used as incentive mechanisms in peer-to-peer networks. In incentive solutions, e.g. [82, 83, 84, 85, 86], information on peers contribution is collected in distributed or centralized manner and peers receive services from other peers according to their previous contributions.

In home environments, behavior monitoring has been mainly used only in networks, which are managed by skilled administrators. However, in these cases the monitoring has been a reactive tool enabling detection of ongoing intrusions and attacks. As the home networks become more complex, there is a need for authorization solutions, which grant permissions according to past behavior and reputation of device or software (either a particular software instance within a particular device or a multiplication of a particular software product).

4.2.3.4 Context-awareness

The term of context-aware computing was introduced by Schilit et al. [121]. Context-aware behaviour can be used to make security both easier to use as well as stronger.

Covington et al. [8] defined a generalized role based access control model. The model enhanced role-based access control (RBAC) by defining a concept of environment role. Environment roles are activated in particular situations. They define which user (subject) roles can access particular resources (objects) at that situation. For instance, there may be roles called 'high CPU load', 'Monday afternoons' and 'downstairs'. Zhang et al. [122] extended RBAC model so that contextual role assignments and permission assignments of particular user are adjusted dynamically. Ko et al. [9] proposed an approach for presenting context-aware access control policies with semantic information. An access is allowed if the request context is semantically equivalent to the context specified in the policy rule. Toninelli et al. [10] presented a context-aware policy model where context is any characterizing information about controlled system entities and about their surrounding world relevant for enabling entities to operate on resources.

An example of context aware authorization is the case where any user in a living room is allowed to control home theatre equipment without authentication, whereas a remote user may be required strong authentication and permissions before allowed to access the same equipment. In addition to context of users (i.e. subjects of access control), also the context of services (objects of access control) may change. The user, who is in a phone or watching a movie, may want to be unavailable for other communication requests and that a tracking service does not reveal location information to everybody.

4.2.3.5 End-to-end Authorization

The authentication and authorization solutions in existing protocols and frameworks control how two devices or a user and a device can interact. Situations where a service is accessed through other devices are controlled only by requiring and trusting these middle devices to control access accordingly. However, for an individual device it is difficult to know what it is allowed to do on behalf of another device. Therefore, there is a need for solutions where authorization and authentication for the whole end-to-end interaction is managed and controlled.

The ***permission attenuation*** concept can be used to model how cooperation should affect to authorization permission. It provides a unifying model for managing

credentials of all affected participants, not just one particular participant such as end-user or user's device. The concept is independent of the implementation or decision making architecture i.e. the model can be implemented in distributed or centralized manner.

Services may operate on the behalf of users or other services. Also, the users or programs may access services using different devices, which provide different security levels. Therefore, it is not enough to authorize just devices or programs, which makes service requests, or users, who initiate use of services. Instead, all entities participating to service request must be trusted and have authorizations to use services.

Permissions, which are available for an access control subject, depend on the session the subject is on. When using intermediate services or mechanisms, which are fully trusted, only a subset of permissions is available for the subject. Actual permissions are the cross section between the permissions groups that individual participants have.

Permissions achieved or permission limitations due to contextual situation should be considered separately for each participant. In ideal case all authorization credentials for all kinds of elements (including users, programs and devices) would be defined in the same consistent way, the likehood of configuration errors and, thus, security holes is mitigated.

4.3 OpenHouse – Secure Platform for Home Services

This subsection contributes by proposing of lightweight, non-centralized access control system for networked home devices. The proposed platform, originally presented in Article III, is called OpenHouse. The subsection focuses on the use of existing widely adopted open communication protocols and on the integration of legacy equipment with home networks. The proposed platform enhances widely used standard for home networking, Universal Plug and Play (UPnP), with TLS authentication as well as with role and domain based authorization. This makes it possible for any networked home device to really know who is trying to access and control it without requiring the end-user neither to log in with user name and password nor to make complicated configurations. The subsection investigates how to minimize the impact of adopting this

approach for both the end-users and developers and suggest areas in which further standardization or guidelines would help. Finally, in order to verify the feasibility of the access control system, the proposed system has been implemented to small embedded devices and its performance has been measured.

4.3.1 Access Control based on User Roles and Certified Service Domains

This subsection describes the design of the access control solution in OpenHouse. First, we describe two enabling building blocks, namely the authorization model and the authentication mechanism. Then, we discuss what is the user impact i.e. what configuration is required from the end-users.

4.3.1.1 Fine-Grained Authorization Model

To ease the management of access control, OpenHouse adopts an approach where users and resources are grouped in a security relevant manner. The selected grouping scheme is the role-based access control model (RBAC), presented by Ferraiolo et al. in [97], which has been extended with domain-based resource classification, utilized in domain and type enforcement (DTE) model [116, 117]. An advantage of RBAC is that new users can be given already defined, preferably intuitively named, roles and, hence, all security policies for a new user can be specified with a single operation. Correspondingly, resources, which are similar, can be grouped and particular users can be given access to grouped resources with one operation. It is practical to do this grouping of users when the amount of users is large. Similarly, it is feasible to group the resources, when there are a large amount of similar resources.

The authorization model, which we adopted, is illustrated in Figure 1. Access control subjects - either end-users, programs or devices - are given roles according to the RBAC model. Similarly, access control objects - services, devices, or data entities - and actions related to them are classified to domains and domain actions.

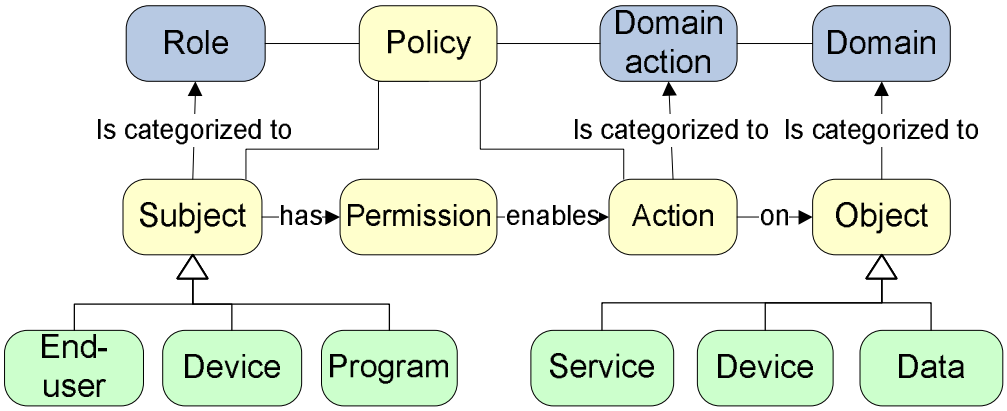

Figure 22. Role and domain based authorization model [Article III]

Examples of roles and domains are given in Table 1. Domain actions are generalized actions available for particular domains. They may describe capabilities e.g. to modify, add, remove, or query object. Subjects' permission to perform actions on objects are defined with authorization policies. Policies link also roles to domain actions and particular actions. These policies specify which actions and service domains are available for which particular users and roles. A typical policy entry for instance would be to say that only users with the role of parent are allowed access to services or sections of services marked with the domain "parental control".

Table 1. Examples of roles and service domains

Role examples
Parent
Guest
Child
Administrative device
Service provider
Shared device

Service domain examples
Personal
Private
Parental control
Security / safety sensitive
Digital rights management
Shared service

4.3.1.2 Security Certification Ecosystem

The presented access control model assumes that users, devices, programs, and services are classified in security relevant manner. However, this kind of classification may be difficult for typical residents. Therefore, we propose an alternative model where part of the categorization can be done by trusted external parties. This certification-based security approach is similar to the certification systems, where programs or devices are

checked and certified by a trusted third-party before delivering for the consumers; such as Apple AppStore, Microsoft MarketPlace, or Symbian Signed.

Potential actors and the phases involved in the certification and access control of networked home environments, and the relationships between them, are illustrated in Figure 23. Service developers are responsible for classifying services using some standardized approach. Different standards including e.g. UPnP and Bluetooth provide already now service classes, interfaces or profiles, which could be utilized when making authorization decisions. However, existing service classifications have not been made from a point of view of security. UPnP forum could be a potential standardization body for defining security relevant domains, which the service developers must use at the service development phase. However, in practice getting security classification to an established standard like UPnP might be challenging. The more realistic scenario might be that there would be a third-party classifier. The third-party could be a commercial service provider or an open community, which the user trusts. This approach requires that homes are able to identify services in trustworthy manner. This identity information can then be mapped to service classifications, which are available from the third-party. As an alternative, end-users could classify services, in the service deployment phase.

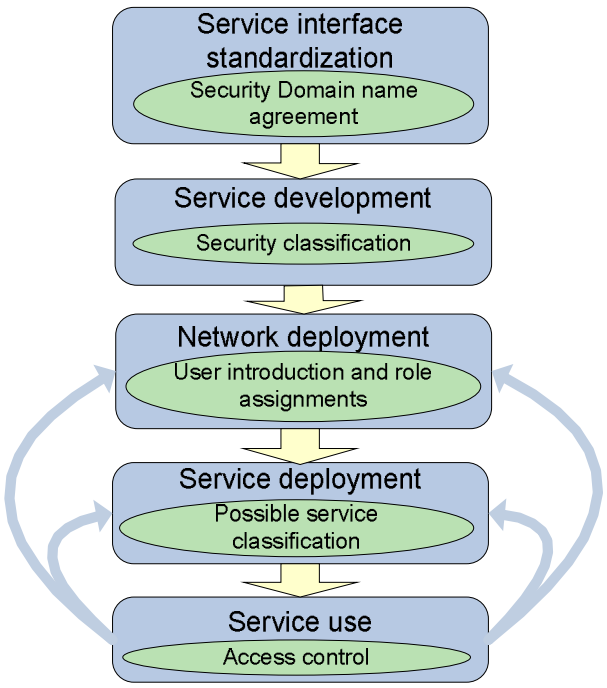

Figure 23. Phases and security tasks in service development and deployment [Article III]

End-users are responsible only for making role assignments as well as for providing security policies, when introducing new users or devices in the network deployment phase. A third-party service provider such as their broadband provider may provide default policies for homes so that end-users are not required to define them. After these classifications the access control is operational. The residents are required to make simple authorization related operations only when introducing or removing users, devices, or software or when changing high-level policies (see Subsection 4.3.1.4 for discussion on end-users role and experience).

4.3.1.3 Authorization Architecture based on Certified Roles and Domains and ACLs

The general model proposed above can be enforced with different kinds of security architectures. For OpenHouse, we adopted a solution which uses both ACL and certified attributes (A4 class in Figure 21). The solution was named as *role authentication*. This approach was suitable for homes where new services and devices are deployed quite regularly. The solution does not require any centralized authorizer component to be always available. Also, in this approach, adding new users and assigning new roles to them is easy and adds very little extra complexity to the out-of-box experience. The approach was designed and implemented using TLS client authentication and X.509 certificates.

TLS enables authentication and enforcement of authorizations. Devices on the home network can mutually authenticate each other by using public keys contained in X.509 certificates. TLS provides several advantages: it is high secure and mature and distributed (there is no need for a centralized server to be available whenever services are used). TLS can be used with different network mechanisms providing TCP/IP including Ethernet, WLANs and power-line protocols. Also, authentication can be extended to users and services outside the home network, i.e. on the Internet, as long as issues related to middleboxes between Internet and the home, such as addressing, are managed.

Each security aware device in OpenHouse has up-to-date information on domains and policies. When clients make service requests, a TLS handshake gets executed and TLS client authentication happens. The client presents an X.509 certificate which contains role information provided as an extended attribute. When an UPnP service receives such an action request, it queries an authorization module, if a client with the role specified in the certificate is allowed to access the service. The authorization module is a logical element which serves to interpret the policy files and could in principle be embedded into the services themselves or be placed running on any device in the home.

TLS and authorization enforcement can be implemented at the system level. This would remove the need from service developers to implement authentication, confidentiality or other security mechanisms. It is enough that they classify their services and make queries to the authorization module when service requests are made. Application protocol stacks in consumer electronics devices must support TLS, which is used to authenticate mutually client and service as well as client's role. From the implementation point of view this means that TLS sockets are used instead of TCP sockets.

The main challenge for TLS client authentication is how to keep the policy and authorization information up-to-date in the different devices on the home network. Basically, there are two approaches. Certificates may have short validity times and, thus, be required to be frequently updated. This solution has the disadvantage that it requires a certificate server to be always available. Also, many devices may not have accurate time information available, so the validity period check may be hard to realize. Alternatively, certificates can be revoked and devices' policy databases updated individually, which may be a manual process. However, to assist users, there needs to be some kind of administrative device, which knows the different devices' security policies and makes updating easier. These devices must be able to create certificates and have sufficient UI capabilities to enable the steps in Figure 24 and Figure 25 to be executed. They must also have access to default security policies. We envisage that the administrative device could be for example a home PC or a smart phone. New devices may be added and certificates provisioned using different association and authentication

methods, which depend e.g. on the hardware capabilities and protocols available in these devices.

When a service request with a revoked certificate occurs, the request is rejected and the client needs to interact with the administrative device to get a new certificate. Such updates are unlikely to be frequent as access control policies in homes are typically quite static.

The solution for provisioning X.509 certificates (or shared keys) at the same time as the new devices are admitted to the WLAN and securely receive the WPA key of the home network was outlined by Kostiainen et al. [123]. The mechanism takes advantage of the fact that the now widely supported new standard for WLAN setup, Wi-Fi Protected Setup (WPS), has placeholders for certificate delivery. It also allows certain devices to be nominated by users as administrative devices (referred to as registrars) and this fits well with our model of administrative devices, which can issue certificates containing role information. WPS is essentially an association method and may also be run after the WPA key has been delivered, making it usable also for the case where a device had a certificate revoked and needs to request a new certificate from one of the administrative devices.

The authorization model can be used to control software component's access to assets inside homes. This is possible when devices are able to control that each software component, hosted in that device, is able to use only its own certificates. Also, services must know how trustworthy each device is i.e. what roles its software components may have. Consequently, the architecture can be used to neutralize attacks of malicious software.

4.3.1.4 User Experience

A key design goal with the OpenHouse solution was to minimize configuration tasks, which the end-user is required to perform. However, some tasks seem to be inevitable. Essentially, there are three situations where configuration may be needed: when adding new devices or users, when deploying services, and when changing policies.

Introducing new controlling devices (e.g. a new smart phone) to the home causes some configuration tasks, which are illustrated with blue trapezoids in Figure 24. The end-user making the configuration may be either the owner of the control device or an administrator, who must specify the owners of the new control device. If the owners are known to the system, the device inherits roles assigned earlier to those users. If the owner is new or when introducing a new user to the network, roles must be assigned for this user. The system may make some additional authorization check when trying to assign roles (such as administrator), which provide access to critical assets.

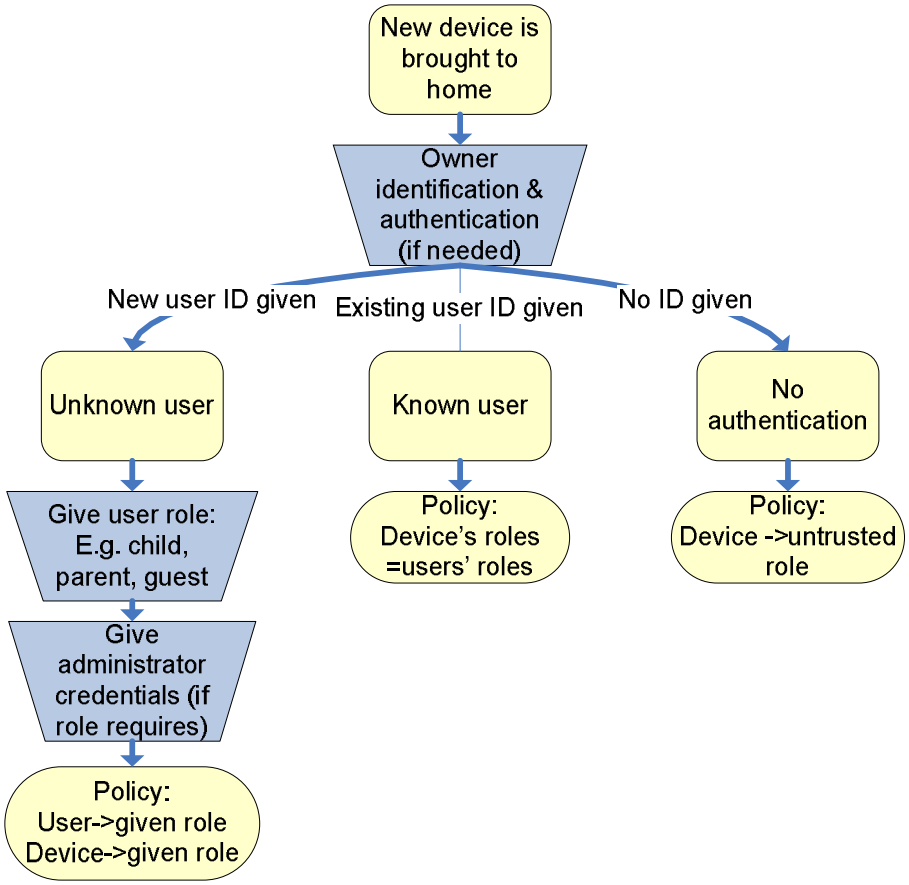

Figure 24. Phases and user actions occurring when adding new terminals/control devices

Introducing new devices, which are hosting services (e.g. a new UPnP based media server), to the home may not require any additional configuration for authorization. If the service developer has classified the service or if any unclassified services are automatically recognized to be part of some particular domain (e.g. like private) no configuration tasks are needed. All that the user must do is to pair the device with the

home network using some device specific key establishment mechanism. If the service is not classified and the home is enforcing strict security policies (which prevent automatic classifications), some configuration tasks, illustrated in Figure 25 will be required. Also, some services may belong to domains, which require additional configuration. For instance, services which are personal in nature may require that the identities of each user who is authorized to access that service are specified. This configuration can be done by selecting authorized users from preconfigured list.

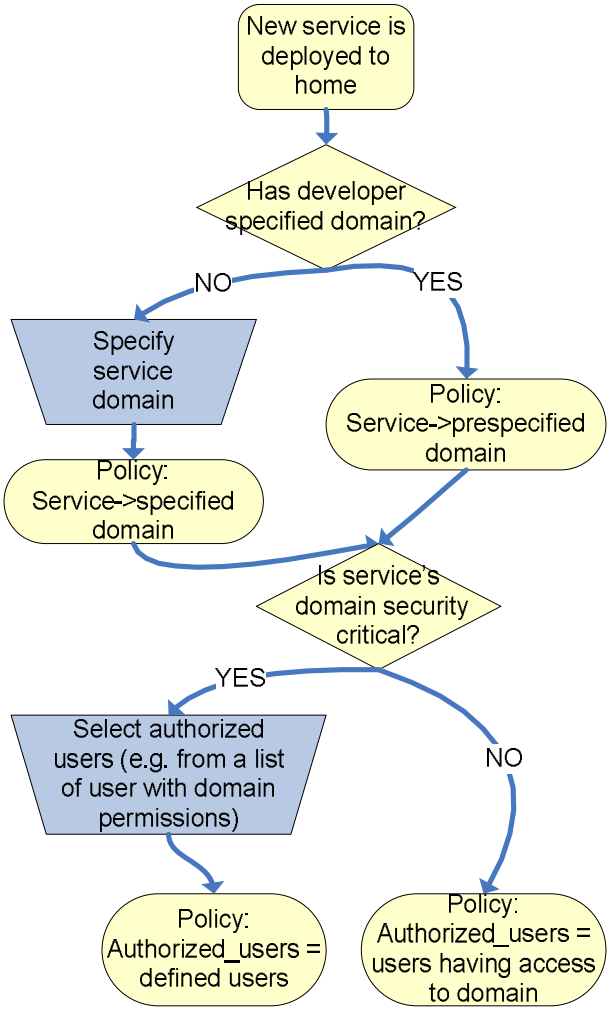

Figure 25. Phases and user actions occurring when adding new services to a home that enforces strict fine-grained security policies

4.3.2 TLS based Security Adapter Implementation for Legacy Devices

This section describes experiences from a prototype implementation. The goal of the prototype implementation was to investigate whether TLS client authentication using certificates would be a suitable technology for the fine-grained authorization model: that it does what we need, and that it is not too slow on the low-end hardware typically used to network legacy home devices. Another goal was to optimize the ease with which the model can be taken into use for developers.

4.3.2.1 Prototype

The prototype implements OpenSSL [124] based authentication layer, CyberLinkC [125] based UPnP stack and a fine-grained authorization module, which UPnP service developers can utilize with minimal effort. The UPnP stack was running on both a Nokia N800 Internet tablet and a small embedded module with Linux and 400MHz processor, called Gumstix.

The service platform was demonstrated by running UPnP thermometer and camera services on the Gumstix platform. We selected Gumstix for this purpose because it is a flexible, low cost platform which can easily connect legacy home devices to network and represent them as UPnP devices and as such is a realistic representation of a consumer electronics device's typical hardware capabilities. The camera was used to take pictures of a legacy digital thermometer. These pictures were analyzed on the Gumstix device with optical character recognition (OCR) software and the resulting temperature reading was sent over Wi-Fi to N800 device's user interface. Pictures were available directly for the N800 device belonging to a parent (i.e. presenting a certificate with parent's role) but not for a user with the guest's role. The demonstration setup is illustrated in Figure 26.

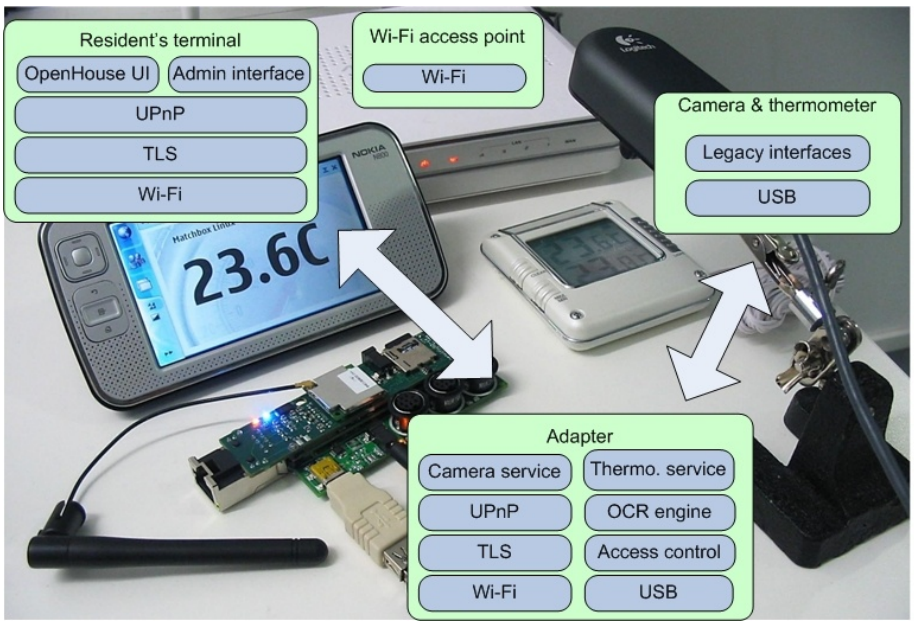

Figure 26. The prototype where pictures from a legacy thermometer are analyzed with a low-end Linux hardware and the temperature results are delivered to Internet tablet device using TLS secured UPnP [Article III]

The client software on the N800 device authenticated itself with X.509 certificates. The client's role, either parent or guest, was carried in the certificate's extended attribute. The Gumstix UPnP device hosted a policy database containing information about which domains the services belong to. Typically, pictures from every camera inside home may contain privacy critical material. Therefore the fetching of camera images was classified as a private service. Sensed information, like temperature, is not typically privacy critical and can be made available for outsiders who are monitoring the home. Therefore, getting thermometer information was classified to be a shared service.

The Gumstix device also maintained applicable access control policies, stating which actions are allowed for which users and roles: parents were given access to all classes whereas guests were given authorization only for shared services. Service developers must add a function call, illustrated in Figure 27 to their source code to enable authorization module to make an access control check.

```
if ( authorization_check (
     get_current_certificate(),
     POLICYFILE, SERVICE_NAME, actionName)
     == AUTHORIZATION_FAILURE )
{
   error_message("Unauthorized request");
   return FALSE;
}
```

Figure 27. Function call, which must be added to UPnP services, for authorization check [Article III]

The above approach required service developers to implement a call to the authorization module. As an alternative, authorization checks could be done completely in the system level. For instance, UPnP protocol stack could resolve target service and action and then make the check. The problem with system level authorization check is that the granularity of the access control suffers as services cannot make data specific checks, which require more understanding of the service than the protocol stack has. However, this system level approach might make the proposal more easily acceptable for service providers and also compatible with legacy services.

4.3.2.2 Performance Evaluation

The feasibility of this usage of TLS authentication was analyzed by measuring the latency between sending a 'get image' request to the camera service and receiving a single packet with a picture. The performance was measured with the UPnP stack without any security features; and with RSA based authentication with or without fine-grained authorization. Further, we studied how the size of the transmitted picture affects performance.

Operation variations were performed 100 times and average values for round time measurements are given in Table 2. In addition to the security protocol, payload size and UPnP messaging, other issues like operating system scheduling, affect the measured performance.

*Table 2. Performance measurements of the prototype **[Article III]***

Message size (bytes) / Protection	1B	1kB	10kB
Unsecured	138ms	187ms	270ms
TLS (RSA)	440ms	504ms	558ms
TLS (RSA) & fine-grained authorization	449ms	522ms	601ms

Experiments show that for individual small UPnP actions TLS authentication causes significant performance penalties. This is because of the heavy TLS handshake protocol: keys as well as message authentication codes must be computed for each call. The penalties caused by the authorization call are relatively small. Furthermore, communication inside the home may not typically consist of individual operations but rather of longer sessions such as frequent fetching of images or media streaming. Hence, TLS handshakes are not needed frequently and the security overhead is smaller.

5 Secure Semantic Technologies for Ubiquitous Network Applications

Semantic web technologies, initially proposed by Berners-Lee [19] and specified by W3C [126], have been seen [127, 128] as a prominent enabler of application level interoperability. This section describes approaches for securing ubiquitous network applications, which are based on the semantic web technologies. The section will reintroduce the vision of smart spaces, where semantic technologies are utilized to enable interoperability in different ubiquitous applications. Then the section will survey security requirements within semantic technologies and smart spaces. After that, the section contributes by describing security architecture and authorization model for smart spaces. The proposed authorization model provides a reusable and interoperable mechanism for fine-grained and context-based access control. The section is based on results implemented in the Sofia project [129, 130] and initially presented in Articles IV, V, and VI.

5.1 The Vision of Smart Spaces

Smart spaces, as illustrated in [131, 132], are physical spaces where information on the environment is collected, shared, and utilized in a context-aware manner. A smart space may be established in different physical environments including, e.g., homes, buildings, vehicles, offices etc. Smart spaces consist of cooperative devices, which are autonomous and able to adapt their behaviour in dynamic manner. The smart spaces are based on context-aware and autonomous computing paradigms as well as semantic web technologies and brokered communication paradigm.

5.1.1 Ubiquitous and Autonomous Computing

Availability of different networked devices, sensors, actuators, and gadgets, have created visions of ubiquitous and pervasive computing. In these visions, described e.g. by Weiser [133, 134] and Satyanarayanan [135], all kinds of computing devices and services become invisible and transparent part of our physical environments. Devices,

which are integrated to buildings, clothing, vehicles, and infrastructure, communicate with each other in order to assist users in everyday living without being noticed.

The transparency of devices is possible only when the devices become autonomous i.e. when devices become able to self-adapt their behaviour without intervention from the end-users. Kephert [15] defined autonomic computing as systems that can manage themselves given the high-level goals from the administrators. They divided the concept of self-management into four functional areas: self-configuration for automatic configuration and introduction of components, self healing for automatic discovery and correction of faults, self-optimisation for automatic monitoring and control over resources for optimal behaviour, and proactive self-protection from security attacks. Autonomous elements are able to monitor their operational context. They adapt their behaviour according to high-level goals and constraints set by administrators. The adaptation is done by processing observed context by using reasoning logic and available reasoning knowledge, which maps the observed information to logic and high-level goals. The adaptation can be based, e.g., on users' or environments' situations i.e. context, such as location or time of day. The principles and potential of context-awareness in computing and authorization were surveyed in Subsection 4.2.3.4.

5.1.2 Realization of Smart Spaces through Semantic Information Brokers and Communication Middleware

Ability for devices to communicate and understand each other is a major challenge for ubiquitous computing and for smart spaces. Particularly, it is difficult to enable interoperability between devices designed for different applications or for different physical spaces. For instance, devices following the users in the pocket or in the car should be able to communicate with sensors and gadgets in homes, public transportation, road infrastructure, shops, hospitals, or parks.

Smart space interoperability has been addressed in the Sofia project [129, 131, 132]. The architecture was selected so that each device is not required to cooperate in the low connectivity level. Instead, the project proposed *middleware,* where devices communicate through information brokers according to the *publish-and-subscribe* [136] paradigm. In the publish-and-subscribe paradigm, the devices first store information to

an intermediate broker. Target of the information is not specified by the source. Instead, devices interested of particular knowledge may request or subscribe information. The intermediate broker will notify subscribed parties when updates emerge. Solution makes architecture suitable for ubiquitous spaces with dynamically appearing and disappearing devices and resources. Secondly, the application level interoperability was facilitated by adopting technologies for semantic web [126], making application level communication protocols faster to develop and reusable.

The smart space architecture in the Sofia project [129] consists of two kinds of dynamic architectural components. Semantic Information Brokers (SIB) provide the access to smart spaces as well as information storage, retrieval and subscription services. Knowledge Processors (KP) join to the smart space and publish and consume information in it. Existing SIB implementations include Smart-M3 [131, 137], ADK [138] and RIBS (RDF Information Base Solution), which was initially introduced in Article IV. KPs are essentially software agents i.e. programs that autonomously cooperate on behalf of the user or another program. Their development is facilitated with middleware layer, which can be implemented in the KP-side as software libraries and which hides the complexity of smart space communication from the application logic. Figure 28 illustrates the connections in smart spaces and how smart spaces can be deployed to different physical spaces consisting of heterogeneous devices.

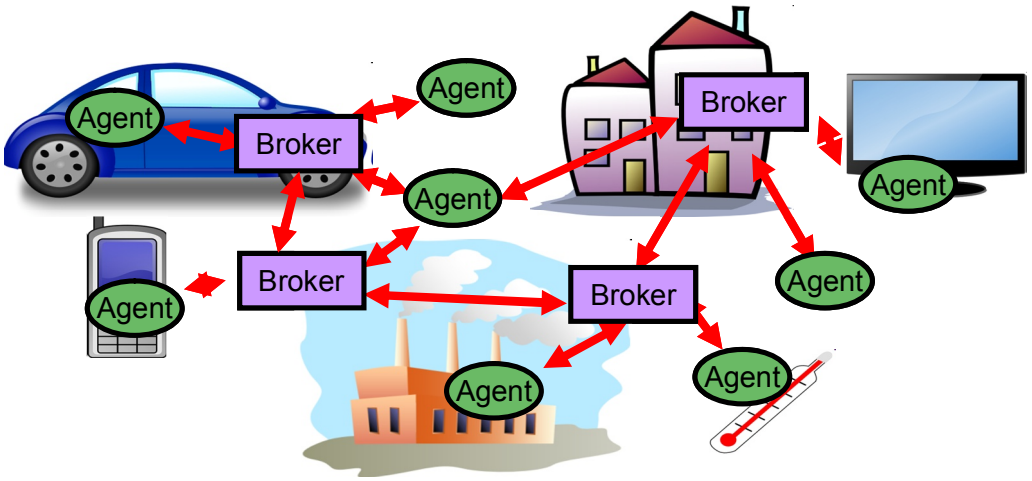

Figure 28. Smart spaces consists of information brokers and application agents, deployed in different physical spaces [139]

Smart spaces have no preference for connectivity mechanism. Any existing communication protocol including Bluetooth (BT), Wireless Fidelity (WiFi) and Internet Protocol (IP) can be utilized. To hide the connectivity specific differences middlelayer communication mechanisms can be utilized. For instance, Device Interconnect Protocol (DIP) [140], which is a key building block of Network over Terminal Architecture (NoTA) , has been utilized in smart space implementations (in Smart-M3 [137] and RIBS [Article IV]). On top of connectivity there is a protocol implementing smart space specific primitives. For instance, the Sofia project [129] has defined Smart Space Access Protocol (SSAP), which defines join, query, update, remove and leave messages and their presentation formats. The SSAP messages are structured using either pure XML or, more compact proprietary, world aligned XML format.

The application-level information is presented and processed using semantic interoperability solutions. The utilized standards include eXtensible Markup Language (XML) [141] for data encoding and Resource Description Framework (RDF) [142, 143] for knowledge representation. To ease information sharing, ontology description languages such as RDF Schema [144], and Web Ontology Language (OWL) [145] are used to define semantic meaning for data, i.e. to define the concepts, properties, and their relations, for different domains. These standards enable applications to use any kind of data models and extend them easily at run time.

On top of the semantic interoperability solutions it is then possible to build smart inference applications, which extract new knowledge from existing information. Inference solutions can be based on various reasoning technologies and application programming models. For instance, answer set programming (ASP) paradigm [146] tackles the heterogeneity related to inference rules in several ways and is a promising approach for reasoning [147, 148].

5.2 Secure Platform for Smart Spaces

5.2.1 Security Requirements in Semantic Web

The technologies for semantic web can be presented using layered models. Berners-Lee [19] presented architecture where Unicode and Uniform Resource Identifiers (URIs) formed the base layer. On top of Unicode and URIs there were layers for XML and xmlschema, for RDF and rdfschema, for ontology vocabulary, for logic, for proof, and, finally, for trust. The left side Figure 29 illustrates a version where the layered model is adapted for smart spaces. The Unicode layer is replaced with inter-device connectivity and the logic layer is replaced with an inference layer. The proof and trust levels are combined into a single trust layer. There exists a large amount of research, standardization work, and implementations of security mechanisms, which can be utilized in smart spaces. The right side of the figure lists the essential security elements, which must be considered when securing smart spaces, and illustrates how these levels map to the levels of the semantic web.

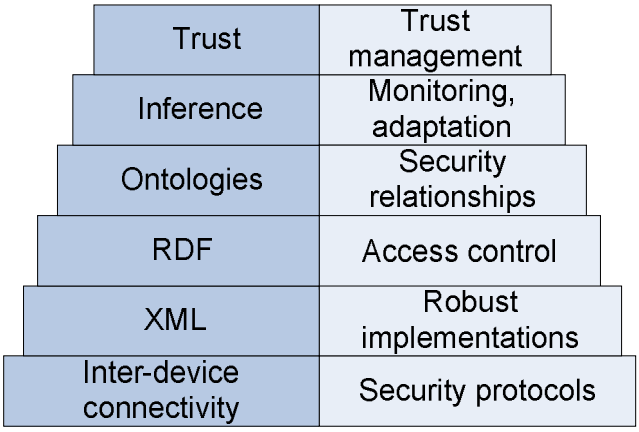

Figure 29. 'Security cake' for smart spaces - Layers of Semantic Web (left; adapted from [19]) and essential security elements (right) [Article VI]

In the following subsections, the security requirements related to these layers are studied in more detail. The subsections also present different scenarios and survey existing solutions.

5.2.1.1 Inter-Device Security

Cryptography, key establishment, key management solutions, as well as security protocols for protecting authenticity and confidentiality of communication and information can be used in the different levels of communication. Security may be in the Smart Space Access Protocol, which KPs use e.g. to join, update, query and subscribe information. Security may be in the connectivity layer, under SSAP. Security may be applied also in the application layer, in which case applications are required to protect data itself.

The use of existing security mechanism to secure smart spaces is not straightforward. As smart spaces are heterogeneous, we cannot assume availability of any particular connectivity-level security mechanism. Security mechanism specified for different communication protocols are not interoperable. These solutions must be able to cope with 'publish-and-subscribe' architecture, resource restrictions and complexity due to dynamic nature of communication. The following paragraphs present essential security challenges, which the developers must consider when designing platforms for smart spaces, namely the heterogeneity and dynamicy.

Security functions are dependent on secure key establishment and deployment mechanisms. Devices must acquire keys and credentials, which enable them to prove their trustworthiness and authorizations for other peers and verify trustworthiness of others. When a smart space supports various security protocols, we need to deliver different kinds of credentials. Also, as physical spaces are heterogeneous it is not likely that single credential deployment model is sufficient. The following scenarios illustrate the different requirements.

Scenario A – *Shared secret for public key certificates*. Devices or KPs, with more processing capacity, may establish session keys using certificates and private keys. These certificates can be requested from certifier, which all parties trust. One approach to control that keys are delivered to correct parties is to protect certificate requests and deliveries with pre-shared secret. This is straightforward approach with some usability and security constrains related to delivery and length of the unique secret.

Scenario B *– Out-of-band models for symmetric credentials and low cost security.* Low resource devices may not be able to secure communication with private – public key pairs. One approach is to deliver symmetric network keys using trusted out of band channels such as Near Field Communication (NFC) or Universal Serial Bus (USB). Some out-of-band models are bi-directional and some one-directional, which will further complicate the deployment. In smart spaces, trusted brokers may forward device specific keys to other devices. The key exchange may need to be controlled by security authorities and forwarding needs to be controlled by users. Further, when forwarding credentials to other devices, we need to consider trust issues. However, devices without direct security relationships may not know how trustworthy mechanisms have been used when keys where initially deployed to the broker.

Scenario C *– End-user specific secrets for access from shared devices.* End-users may use shared or borrowed devices to access data. In these cases we cannot assume availability of existing credentials in devices. It should be possible for users to use e.g. passwords, biometrics or security tokens to access data.

Smart spaces are dynamic. Users, devices and brokers may join and leave at any time. Therefore, spaces should not assume availability of any component. Further, in some smart spaces there may be multiple distrusting authorities. These authorities may control same SIBs and want to ensure that only those devices, which are certified by them, can access shared information. For example, buildings may have devices, which are shared by several families, and malls may have devices used to serve different shops. Therefore, smart space platforms should provide solutions for adding new authorities. Since also authorities may emerge any time, these mechanisms should be dynamic and preferably not involve actions from SIB provider. Mechanisms should also be provided to enable users to determine trustworthiness of authorities.

5.2.1.2 XML Security and Robustness

Vulnerabilities in software implementations, particularly in those which are processing and parsing input and XML documents, have been a major source of security problems in Internet. In smart spaces with embedded devices this issue is even more critical as these devices may not have direct Internet access, which could be used for security

updates. Hence, robustness of software implementations must be in the focus from the start of development. Robustness against malformed content can be achieved with careful coding practices as well as mature and security tested interfaces.

In the XML level, there are solutions for protecting integrity [149] and confidentiality [150] of XML documents. There are also solutions for defining access restrictions to elements of XML documents. There is a prominent standard from W3C, XML based eXtensible Access Control Markup Language (XACML) [151]. XACML standardizes XML notations to describe the authorization policies.

5.2.1.3 RDF Access Control

Smart spaces are vulnerable to various confidentiality and privacy related threats, which must be addressed with access control solutions. To illustrate the requirements, consider a scenario where the user makes a physician appointment with mobile phone and gets a confirmation with time and address as a text message. This information is then used in smart spaces at the home, at a car, at the hospital by different applications including calendar, navigator or elevator controller. The information must be protected so that details of appointment are available only for the user itself. Time and destination may be available for the family and navigator. For elevator, which is in public smart space, only the destination floor is revealed. Therefore, we need solutions for protecting authenticity and confidentiality of communication as well as for controlling access to information. These solutions should fulfil smart space specific requirements when considering security level, complexity, required implementation efforts, maintenance work and performance. Solutions should be applicable for embedded devices with limited communication, processing, memory and battery capacities. Also, solutions should work in dynamic environments where new devices may join, store and subscribe information and leave at any time.

Information in smart spaces is stored in RDF format. RDF is a data modelling approach where statements are made of resources. Statements are made using subject-predicate-object triplets. Collections of triplets form directed graphs. The subject and object refer to resources (nodes in graphs) while the predicate refers to the aspect of a subject and defines a relationships between two resources (edges in graphs). Object resource may be

either a literal or Uniform Resource Identifier (URI) and subject resource is either a URI or unnamed empty branch.

SIB is responsible of authorize access to every RDF resource. Different strategies to store access control information to RDF database are possible. A straightforward approach is to tie authorization policy directly to each RDF resource under protection. An example of RDF resource specific policies is illustrated in Figure 30. RDF does not enable direct links to be added to literal nodes. Therefore, Policy3 in the figure cannot be presented with RDF. This means that we must have some alternative mechanisms to protect specifically literals or that we accept this limitation in the granularity of protection (and protect all literals under particular RDF branch using the same policy).

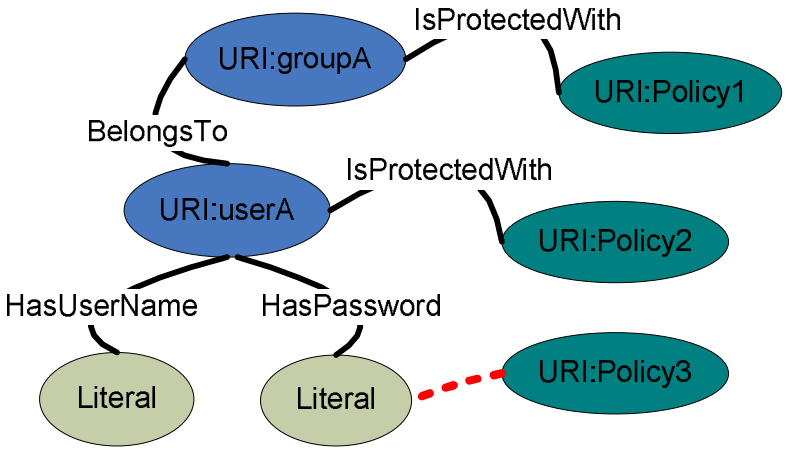

Figure 30. Example of RDF graph and alternatives for embedding access control policies [Article VI]

To control access to shared semantic information, various fine-grained authorization models have been introduced for RDF. These approaches include approaches were access control is implemented as an additional layer on top of the RDF repository, as in [152], and approaches where access control information has also been integrated into RDF repositories. In the triple-level access control [153] RDF resources are protected with access restriction properties. Essentially, these properties are links to RDF access policy graphs that specify the owner of RDF resource as well as those predicates that this protection applies. Permission assignment mechanisms in smart spaces must be self-managing as devices or users may join or leave smart spaces at any time. Since new users should be able to access existing data, they must be provided sufficient credentials

at any time. Models where end-users define access control policies explicitly for each RDF resource become infeasible when the amount of information and devices increases. Therefore, some researchers have proposed models where RDF-level access control decisions are implicitly derived from existing higher-level policies and context information. In [154] a policy-based access control model is presented enabling metadata to be used when defining permit or prohibit conditions. Also in [155] a RDF class hierarchy is utilized to manage and derive access control policies. [156] proposes a high-level policy specification language for annotation RDF triples with access control information. Moreover, approaches for access control reasoning based on concepts and their relations represented by ontologies have been introduced by [157] and [158].

However, semantic reasoning for real time security control is a challenging task when the size of ontologies and information grows [159, 160]. Consequently, to enable real-time security enforcement with expressive and complex ontologies, efficient and scalable solutions are needed. [161] addressed scalability issue by limiting the granularity of the access control. Their model, targeted for clouds, used RDF graph elements as user permission tokens. In Subsection 5.3.2, a simple RDF resource level access control model for optimized RDF information broker solution is presented.

5.2.1.4 Ontologies and Security

Ontology can be defined [162] as a shared knowledge standard or knowledge model, which defines primitive concepts, relations, rules and their instances. Hence, ontologies can be used to define concepts for security data, policies and security relationships. This information can then be used in smart spaces to select appropriate protection for different types of information. Ontologies are needed because it is not always feasible for KPs to explicitly store security data and policies, which is the case e.g. when information is generated within inference layer.

Some general ontologies, which are targeted for smart spaces, such as Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) [163], have adopted elements for defining access control policies. Further, other ontologies used within smart spaces may be extended with ontologies defining security concepts or access control rules. Ontologies, which define information security concepts, include e.g.

Ontology of Information Security (OIS) presented by Herzog et al. [164]. This ontology was extended in Information Security Measuring Ontology (ISMO) by Evesti et al. [165] with intergrated metric related concepts for classifying and quantifying security levels of security solutions.

5.2.1.5 Security Monitoring and Adaptation

In a smart space, several KPs may insert and remove information. A KP can subscribe information changes and when the information changes the KP can make further changes as well as other activities. These other activities form the base of the physical smart space behavior that is experienced by people and sensing KPs. In order to avoid chaos in smart space, activities behind physical behavior need to be aligned. Action level interoperability may require a higher level plan for smart space as well as a mechanism for detecting and eliminating the effect of misbehaving KPs.

Inference techniques may be used to infer security information from existing knowledge (i.e. RDF data presented in a form of ontologies). This new information can then be utilized to adjust or adapt systems security behaviour. For instance, RDF level access control can be adapted according to infererred information. Inference can also be used as to detect inconsistencies within data. The correct actions in various information security situations are application dependent. Inferencing can be based on different reasoning engines and programming or rule languages. Existing solutions have been surveyed in different benchmarking studies, such as [159, 160].

In smart spaces, semantic reasoning has been used to resolve different application specific questions. For instance, Answer Set Programming (ASP) paradigm has been used to solve resource allocation and deadlock activities [166]. In addition to application specific security situations, there are some generic information which can be monitored. Particularly, it is possible to monitor information producers, i.e. KPs authoring information, and consumers, who form an audience for the information. A broad audience may indicate the importance of the information. Also the smart space information is useless if it does not have any consumers or producers.

5.2.1.6 Trust Management

Trust in and for smart space gradually increases when it operates according to a plan and when deviations from plans do not cause negative effects.

Trust in smart spaces can be considered in three basic levels. The first level is the trust towards technical robustness and reliability between KP and SIB cooperation and security mechanisms. All layers below trust layer in Figure 29 build this trust. Mechanisms and methods used in each layer need to guarantee both correct operation and robustness in case of misuse and exceptions. The security aspects need to be considered and embedded into each layer. The second level is the trust between KP and SIB as well as the trust between the end user and SIB provider. This level addresses question what information SIB is trusted to guard and which users are trusted to access smart space. The third level of trust emerge between different KPs and end users in the smart space.

Trust for correctness of the information is based on trust to the origin of information. Different smart space users and devices may be trusted to perform different actions. This trust may be based on directly monitored behaviour or on certification by a trusted party. Trust information describing whether peers handle data according to expectations and trust that peer does not behave maliciously should be stored in SIB. This information should also be delivered for smart space devices so that they can adapt their cooperation according to peers' trustworthiness.

5.2.2 Security Architecture for Smart Spaces

This subsection proposes security architecture for smart spaces, presented initially in [Article IV]. The architecture integrates solutions for protecting confidentiality and authenticity of information exchange. Furthermore, the architecture enables remote monitoring and control of systems security state as well as fine-grained authorizations over smart space and information access.

The proposed architecture is illustrated in Figure 31. The key component in the architecture is Semantic Information Broker (SIB). SIB brokers and protects information produced by knowledge processors (KP). The figure illustrates two separate roles for KPs: information producer and information consumer. Additionally, there are

specialized software entities (KP roles) for administrating and for monitoring security. In the figure, security relationships between different actors in smart space are illustrated with light blue arrows. Blue arrows illustrate actual information flow. Green ovals illustrate the key security information, which is exchanged between devices and discussed more closely later (C=credentials, SP=security policies, TD= trust data, and KP_ID=identity of information producer).

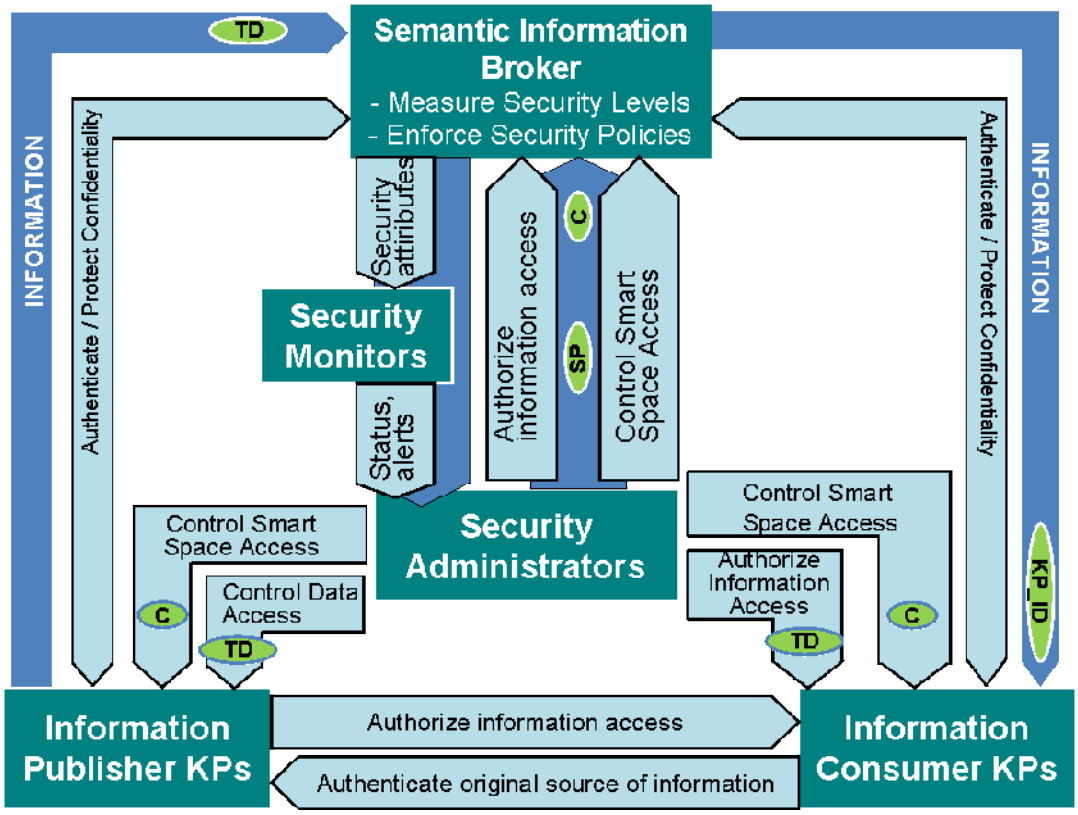

Figure 31. Smart space security architecture [Article IV]

Security requirements between KPs are the following. The producer KP needs to control which consumer KPs can access the information it produces and consumer KPs need to authenticate the source of information. Smart spaces may also have administrators, e.g. person owning the SIB device, which may set some particular requirements on who can access smart space and how information in smart space can be shared.

Authorization related requirements are enforced in SIB, which controls who can access which piece of information. This control is done according to security police directives (SPs in the figure) coming form the administrator. The illustrated architecture is logical.

Typically, security administrator's role may be divided to several devices and KPs. For instance, KPs may independently control who can access the information they produce.

To enable authentication between SIB and KPs, authentication credentials (C in the figure) must be distributed among smart space participants. Administrators may also distribute additional trust information for controlling security and trust issues within smart spaces.

There is no built-in end-to-end authentication protocol. Authentication between KPs is based on trusting SIB to keep track of identities of information publishers and to provide this information for consumers (KP_ID in the figure).

The proposed credential deployment architecture utilizes RDF information sharing mechanisms available in smart spaces. The architecture consists of three components: KPs (or device or end-user, wishing to access smart space), SIBs (relaying credential information), and security administrators (SAs). The solution enables that credentials are deployed through a SIB or directly from an SA to a KP. The direct communication paradigm does not follow the principles of brokered smart spaces communication but may be practical in some situations due to security, usability or cost reasons as explained in Section 2. Hence, in some devices KP functionality is extended with software enabling it to communicate directly with SA devices. The main steps of the proposed protocol are given in Figure 32.

1 *SA and KP establish a shared secret.* SA may also deliver credentials (e.g. X.509 root certificates) which enable KP to verify SIBs trustworthiness. Shared secret can be established using various mechanisms (see Section 2 for some standardized examples). Some end-user contribution is required. The communication may happen directly between KP and SA devices or through SIB.

2 KP registers itself to SIB

2.1 KP creates requests for each technology specific credential it requires. The request is encrypted with the shared secret and contains identity information. E.g. in case X.509 certificates are requested, certificate's name and public key is stored to SIB. In case of username-password pair, KP stores either the username or the pair. In case of symmetric encryption, only device ID needs to be stored.

2.2 KP stores credential request (e.g. certificate requests) to SIB. SIB notifies those SAs, which have subscribed information on new credential requests.

3	*SA provides credentials for KP through SIB and sets access control policies*
3.1	SA decrypts requests and generates credentials. Information on how shared secret was established as well as optional trust information is stored to credentials (e.g. to X.509 certificate's subject name or alternative name fields). SA encrypts credentials with shared secret and stores them to SIB. KP is notified.
3.2	SA modifies KP's information in SIB so that KP gains appropriate access permissions in SIB. For instance, KP may be added to particular groups or given particular roles. Permission assignment is based on information, which can be collected in step 1 (e.g. by SA querying end-user what roles are given for KP). User information is made accessible only for the KP and SA. See 5.2.4.1 for examples of this information.
3.3	KP downloads credentials from SIB and decrypts them
3.4	KP may upload credentials enabling other KPs to interoperate with it directly. These credentials are protected by setting appropriate access control policies.

Figure 32. Credential deployment protocol

An example of architecture with multiple authorities is illustrated in Figure 33. In the figure blue arrow (1) illustrates key establishment, red arrow (2) KP registration and black arrows (3) credential delivery.

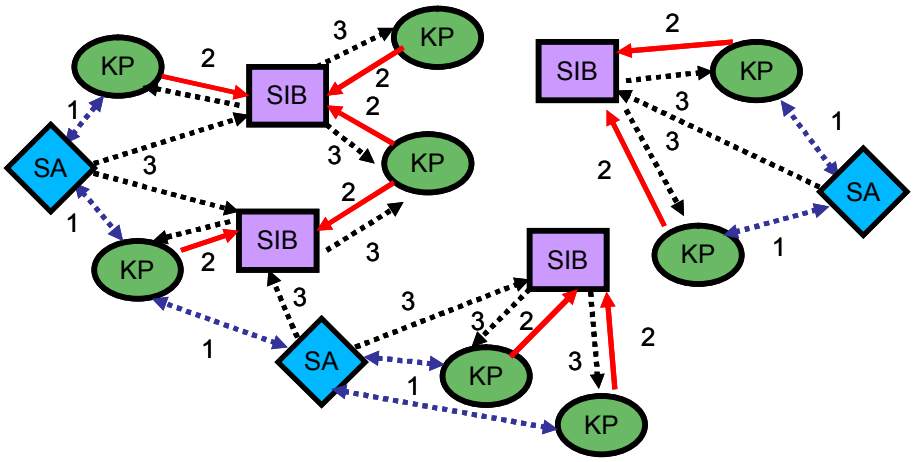

Figure 33. Smart space architecture example with brokers (SIB), knowledge processors (KPs) and multiple security authoritys (SAs) [Article V]

The advantage of this indirect credential establishment model is that SIB can distribute any kind of credentials. Hence, if we have e.g. created security session with Bluetooth, the end-user is not required to perform any more actions in order to use also TLS or wireless local area network security within the same space. Also, a KP can get credentials, which enable device to directly contact other devices in smart space.

Further, KPs may use this same approach to renew existing credentials. Of course, it is possible to deliver credentials and some permissions, directly at step 1, without the overhead of step 3. However, when SA sets KP's security attributes directly to SIB it can more flexibly control KP's permissions. E.g. by modifying role assignment it can add and revoke some permissions without revoking the whole credential.

The secret established in the step 1 is used for protection against man-in-the-middle attacks. Security administrator may use phase 1 also to inject trust information within smart space devices. Trust information refers to any additional attribute information that describes KP's trustworthiness. For instance, the strength of this established secret depends on the method that was used to establish it. If possible, the credentials will contain information identifying the credential delivery method. This trust information is later on used when making authorization decisions. For example, if Bluetooth pairing mechanism is considered to be weak, the KP cannot use the TLS session to gain access to critical information. In more general, any static trust information may be embedded to credentials. This provides an efficient way to control which users can be allowed to access critical information. For instance, we may define access control policies, which restrict data access from users with particular security level.

The model provides flexibility as it enables that SA does not need to be available when KP registers to SIB, SIBs do not need to be available when KP and SA make connection, and KP does not have to be available when credentials have been created. Access to SIB is gained when it becomes available. More permissions for KP are gained when also SA joins the smart space.

SIBs can enforce policies coming from devices, which have been certified by different SAs, this is needed to keep certification process lighter (as one authority does not need to do all operations), and more secure (as authorities do not need to trust each other). Before the credential deployment is possible, an SA and a SIB must have established a trust relation. In this phase, the SA delivers credentials (e.g. X.509 root certificates or shared secrets), enabling SIB to verify KPs, which will be certified by this SA.

5.2.3 Secure Smart Space Communication

This section describes our security implementations for a smart space platform. The security has been implemented to several components in RIBS communication software stack and to security KPs as illustrated in Figure 34. Dark green components in the top right illustrate security components for KPs and three components in the opposite left provide access control enforcement for RIBS.

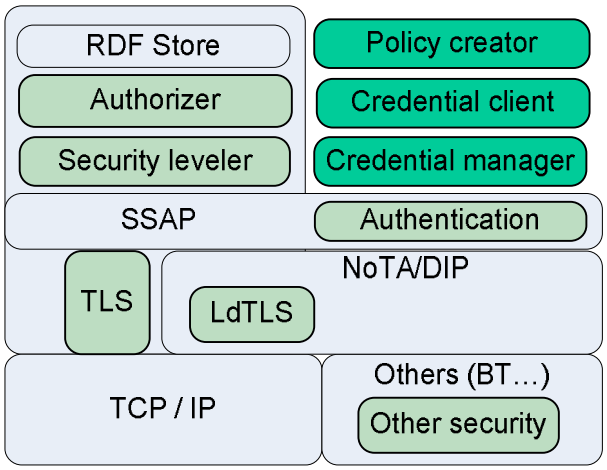

Figure 34. RIBS communication software stack and security components [Article V]

The lowest layer of the figure contains connectivity alternatives, which include TCP/IP protocols as well as e.g. Bluetooth. These connectivities may have own security protocols but they are not assumed to be secure. Connectivity alternatives are used by Smart Space Access Protocol (SSAP) directly or through Device Interconnect Protocol (DIP). Communication security is achieved with Smart Space Access Protocol with Transmission Layer Security (TLS), Device Interconnect Protocol with TLS (DIP/LdTLS) or with connectivity (such as Bluetooth) specific mechanisms. There is also a end-user authentication solution in the SSAP layer. Security leveler and authorizer components are used to control, which users are allowed to access RDF store. Additionally, we need programs for controlling access control policies and for credential management.

The basic security operations within RIBS occur as follows. When a KP joins smart space, by sending a join request to RIBS, credentials and information on security parameters (communication protocol, ciphers, credential deployment mechanisms etc.)

are passed for the security leveler and authorizer components. The leveler normalizes security parameters so that authorizer may use information from different security components when controlling access to RDF store. Authorizer makes fine-grained access control decisions on the RDF node level. For each node, it is possible to define different security policies, stating e.g. which users are allowed to read data or who is the authority. Additionally, it is possible to set requirements for the security strength level or trustworthiness of the KP or communication session.

Security policies are stored by any user, who has sufficient permissions to do so. For storing policies and for introducing users, there reference implementation for creating policies and for managing users credentials. As policies are sent as RDF triplets RIBS, no security specific mechanism for policy delivery or storage is needed. Policies can be set explicitly for each node or they can be implicitly derived using ontologies as described in the following sections.

5.2.3.1 TLS Adaptation for Device Interconnect Protocol

Device Interconnect Protocol (DIP) [140] is a middleware communication solution. DIP provides consistent socket API for application developers and hides protocol details such as addresses. DIP implementations provide adapters for different transport protocols such as TCP/IP and Bluetooth.

The security has been implemented as a new TLS adapter (named LdTLS), which provides a security solution for connection oriented communication. LdTLS uses OpenSSL library's TLS/SSL protocol implementation to encrypt and authenticate TCP/IP communication. The implementation is an extension to LdTCP module. In LdTLS, TCP operations have been replaced with TLS operations. When there are several adapters build to the stack, LdTLS is selected by setting priorities. Deployment of credentials and TLS specific parameters from the applications to the LdTLS and other adapters through the DIP stack was enabled by extending the address structure with credentials and by providing a socket option call for delivering security information to adapters.

5.2.3.2 TLS Reference Security for Smart Space Access Protocol

The Smart Space Access Protocol (SSAP) [137] is the protocol for join, leave, update, query and subscribe messages used in RIBS. When a KP joins to smart space, TLS sessions are negotiated and then all communication is routed to TLS sockets. Security based on the connectivity-level solution brings some advantages when considering development efforts, reliability and resource consumption. The TLS protocol can be considered robust due to its wide acceptance and availability.

TLS provides solutions to most identified needs except for non repudiatable KP-to-KP authentication. TLS provides mutual authentication between KP and RIBS. As security connections are only between KP and brokers, consuming KPs must trust brokers to authenticate sources of information. TLS handshake is sufficient for authenticating the KP for the SIB and the SIB for the KP. TLS authenticates that the peer is owner of the certificate. In case we have devices, where single certificate is shared by many users or programs, X.509 certificate based TLS authentication may not be sufficient. Certificate based authentication may also be unfeasible for low-resource devices, which need either lighter authentication mechanisms. Therefore, RIBS may also authenticate end-users using credentials, e.g. username – password combination, which are send in the credential field in the SSAP join messages. TLS is used as a source of trust for end user authentication if authentication procedures are done within TLS session.

TLS implementation supports both GnuTLS [168] and OpenSSL [124] libraries.

Security sessions are kept alive as long as possible i.e. until a KP leaves the smart space or unsubscribes smart space information updates. This connection oriented messaging minimizes the amount of heavy handshake procedures.

5.2.4 Level-based Authorization for Controlled Information Sharing over Heterogeneous Connectivity

Different security characteristics within smart space devices cause an interoperability challenge. Even though the first smart space security solutions, introduced above, were based on TLS, the security approach is not limited to any specific security mechanism. Hence, any key management scheme, pairing model, security protocol, or encryption

algorithm could be integrated to the system. In the presented implementation TLS was integrated to the middleware layer as the availability of security in the connectivity layer cannot be assumed. Smart space deployments may utilize this layer or may opt to use some another customized security solution.

When a KP joins to a smart space or subscribes to particular information, it negotiates a security session with RIBS. During this handshake both parties verify that the peer has valid smart space credentials. RIBS resolves also security context information related to the security sessions and incorporated to credential information. This security context information is then utilized to enforce that the communication session fulfils the minimum security level requirements set for the smart space as well as to authorize access requests to particular information pieces. Security properties and state resolved from KP during smart space join operation can also be stored and published for monitoring applications. These features may also be disabled in order to protect privacy of publishers.

The *security level* is a measurement describing the strength of different security attributes. RIBS derives security level information from TLS session and X.509 certificates. Factors, which are measured, include used security protocol, algorithms and their parameters such as key lengths. Also, X.509 certificates may include attributes such as the strength of (pairing) mechanism used to deliver smart space credentials and platform trustworthiness information.

To illustrate the idea Figure 35 describes how the strength of security protocol can affect to the information flow in smart spaces. In the figure, Device A publishes information and sets a policy that published information must not leak to devices with weaker protection. RIBS evaluates security levels to the joined smart space devices to be 3, 3, and 1, based on their security mechanisms (TLS with AES encryption, Bluetooth security version 2.1, and WPA with unauthenticated pairing, respectively). Consequently, RIBS allows only Device B, with security level 3, to access information.

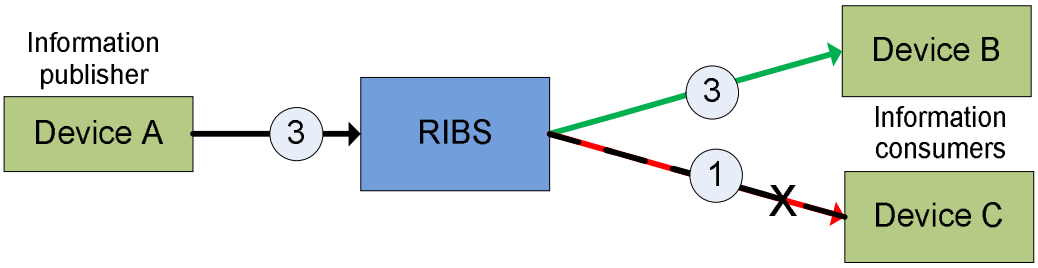

Figure 35. Simple example of security-level based authorization

5.2.4.1 Profiling Security Levels

RIBS collects information of various methods and algorithms that KPs are using and how they use them. KP specific measurements can be made from following categories:

1. Pairing method – i.e. device's smart space deployment information (i.e. information on what pairing mechanism was used when the device was associated to the smart space and certified). This information is given for KP during smart space association and carried e.g. in X.509 certificates.
2. Authentication – i.e. mechanisms for user and device identification for protecting the authenticity of communication. This mechanism is negotiated during security protocol's handshake.
3. Keying – i.e. the used protocol for changing of the network keys. This mechanism is negotiated during security protocol's handshake.
4. Cipher – i.e. mechanisms for encrypting communication. This mechanism is negotiated during security protocol's handshake.
5. Platforms trust parameters - RIBS may resolve run-time trust information (e.g. OS version, protocol implementations, state of antivirus software etc.) from the requesting KP device. For instance, RIBS may query this information directly from the KP using e.g. Trusted Network Connect protocol. Trust information, which is static in its nature, can be integrated to credentials (e.g. to X.509 certificates) when KPs are first introduced to a smart space by security administrator.

RIBS uses these security measurements to determine security levels for each communication session. The security level can be defined in numeric manner. For example, security properties can be profiled to four levels e.g. 'No-Low-Medium-High' as described in Table 9. Derived final security level, used in authorization, is a 'join'

operation for the security levels of each separate category so that the "the weakest link" will be the final security level.

Evaluating the strength of a particular security mechanism is not straightforward as different kinds of attacks are applicable against different mechanisms and because we cannot predict what attacks are feasible in arbitrary smart space. Therefore, the evaluations are somewhat subjective. They can be, however, adjusted both in time and for each smart space. When new algorithms and implementations are evaluated, existing (e.g. cryptological) analyses and information from vulnerability databases may be utilized.

Table 9. An example of security level classification

Security Level	Description	Matching Security Standards
No	Security is not provided at all	
Low	Pairing methods without authentication, authentication algorithm	BT v2.1 just-connect pairing,
Medium	Authenticated pairing, authentication & encryption based on asymmetrical cryptography	BT v2.0 pairing (PIN based), TLS-DES
High	Authenticated trusted pairing, authentication and encryption (that shall endure two years, e.g.)	WUSB numeric association, BT v2.1. out-of-band pairing, TLS, RSA, AES

For example, consider a case where a device is first paired with security control device using Bluetooth v2.0 pairing with 4-digit PINs (personal identification numbers). The device gets smart space credentials from this control device and can then connect to information broker. The connection to information broker occurs over WiFi and TLS. TLS utilizes RSA (Rivest-Shamir-Adleman) and AES (Advanced Encryption Standard) or 3DES (Triple-Data Encryption Standard*)* algorithms with key lengths 2048 and 128 or 168 bits, respectively. Optional platform trustworthiness checks are not made as they are not required in this smart space. The security level is considered to be medium as the weakest link was Bluetooth pairing with the medium level. If, however, pairing had

utilized more secure WPS (WiFi Protected Setup) in-band model or numeric association of WUSB (Wireless Universal Serial Bus), the final security level would had been high.

The presented coarse and one-dimensional security level example ('No-Low-Medium-High') provides sufficient control over security but is also usable enough for end-users to understand. However, it is possible to define different security levels for different smart spaces. For instance, in a smart space that has no needs for typical end-users to control security, we may have multi-dimensional security levels (e.g. dimensions for strength of authenticity and confidentiality). In the future, it might also be possible to define security levels as a part of a security ontology.

Run-time changes to the security level provide some challenges. When the RIBS wishes to tighten up the security level, it sends the leave indication to KPs. After that a KP must join to the RIBS again. If changes are allowed, a KP querying data cannot know if that data has been inserted by a trusted KP. Also a KP inserting data should be able to know how data is protected and that there won't be changes to the protection level. Therefore, in these cases, RIBS is required to keep track of which information has been stored with a particular security level and protect information accordingly.

5.3 Access Control for Smart Spaces

This section presents flexible and reusable RDF access control model, enabling policies to be based on any context information. First, the section presents a conceptual overview on how RDF level security policies can be generated dynamically at run-time. Then, the security model for RDF and our RIBS based implementation are presented. A short presentation of RDF and its security requirements as well as surveys related access control solutions were provided in Subsection 5.2.1.3.

5.3.1 Dynamic Policy Generation

This subsection presents high-level approach for deriving fine-grained authorization information from available security and context information and high-level user policies. The approach consists of three essential elements, illustrated in Figure 36: knowledge, reasoning applications, and RDF-level authorization policies. The approach

proposes use of high-level policy and context knowledge from the smart space. Smart space applications utilize this knowledge to infer low-level policies to fulfil RDF-level access control matrices.

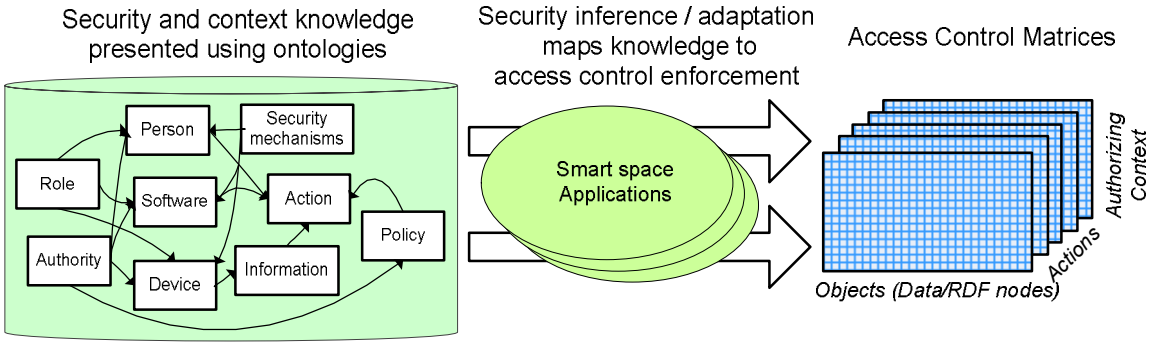

Figure 36. A conceptual model for mapping security knowledge to low-level RDF access control matrices with application specific security inference [Article V]

Smart space applications need to control which users in which context can perform which actions to which pieces of information. Access control enforcer can perform this fine-grained control if the authorization information is available in the access control matrices. Access control matrices [169] characterize the rights of each subject with respect to every object in the system. In smart spaces, this control can be done in the RDF level and be enforced by SIBs. Access control matrices can be presented as a cube, which is a large data structure with information pieces in one dimension, users and context rules in one dimension and actions such as read or write in one dimension. Truth values in matrices then indicate whether action is allowed or denied. In larger environments with multiple users and large amount of data, the size of access control matrices may become large. The management of matrices is challenging if a policy for each RDF resource must be set explicitly.

To ease this configuration, the smart space applications should be more autonomous and able to automatically configure access control matrices. This configuration can be done using available security relevant contextual information and high-level rules. The knowledge is presented using semantic ontologies so that it is easily accessible for KPs in smart space. Security relevant knowledge and their presentations can be based on models, which already exist in the security field. For instance, we can utilize ontologies, presented in Subsection 5.2.1.4. Context information can come from various sensors

and monitoring components, which are available in the smart space. For instance, security level information, presented in Subsection 5.2.4, can be utilized.

RDF-level policies are generated by knowledge processors by inferring policies from the security relevant knowledge. These solutions find whether there are authorizing semantic relationships between the subjects and objects in the RDF access matrices. These applications may utilize programming models and semantic reasoners, as presented in 5.2.1.5. A simple reasoning example could be a user, who has family roles and work assignments. These roles and assignments are related to particular information, which must be available for the user. By scanning existing data rule the solver would notice that there are users with these relations and information whose access is authorized by these relations. The solver can also check that trust rating given for the user fulfils security requirements, which the author of the information has set. Based on this reasoning, the solver can add new entries to access control matrices. The proposed reasoning can be done at the time when information is accessed. Alternatively, to optimize check times reasoning can be done before hand, particularly, when new users are added or when information related to policies is changed (e.g. a security relations related to information is changed).

5.3.2 Reusable Context-based Model for RDF Access Control

Security knowledge presented with ontologies provides means to present security policies, which control behavior of smart space devices and applications. However, analysing and planning access control decisions at runtime, when information is queried and modified, can be computationally costly. In smart spaces the information is shared using SIBs, which are unaware of applications' conceptual policies and hence unable to enforce these policies. SIBs can be assumed to be aware only of minimal set of standard security primitives, which are associated to information elements instead of the meaning of this information. Also, as smart space devices may have limited computing capabilities, solutions based on cryptography are often unfeasible. Therefore, efficient solutions are needed to protect information sharing and to control information access in a fine-grained manner at the level of semantic data.

This subsection generalizes and formalizes the RDF access control approach into a conceptual security model. The model has been verified with RDF but it can be applied to any information presentation system which is based on subject-predicate-object triples. It specifies how access control policies and security control information over resources are structured and presented. Runtime costs are minimized by requiring that each policy is presented with a single information triple. The model is based on context and security measurement concepts, which are used to authorize actions. Hence, the model can be applied efficiently and flexibly in various dynamic security control situations.

Figure 37 depicts the security relationships in the security model. The model has a relation to three software components, presented in the top right corner of the figure. *Smart space applications* insert, query, modify or subscribe information resources. The *access control* component authorizes and controls these operations. Application specific *security adaptation* components administer the behaviour of the access control component. This administering is done by controlling relationships between resources as specified by the security model.

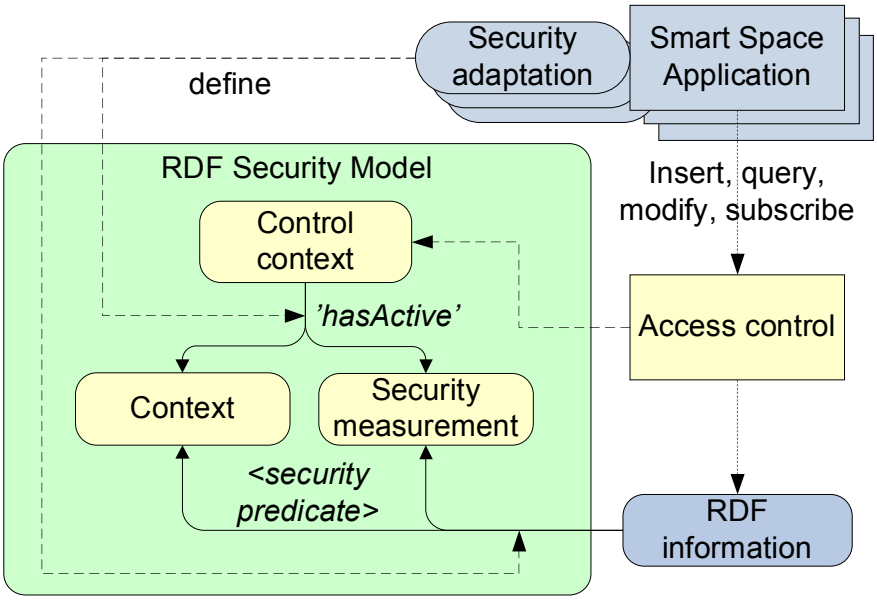

Figure 37. Context based runtime security control model for RDF information [1]

Each piece of information, i.e. each *RDF information* resource, can have a relationship with one or with several *context and security measurement* resources. Each relation presents one access control statement and is described using RDF triplets in a form:

<*Information, SecurityPredicate, Context/Measurement*>. *Security predicates* are RDF properties defining authorizing or accounting relations for the security control. Predicates that are to be used when authorizing RDF transactions are presented in Table 10. Predicates for accounting can be found from Table 11. C*ontext/measurement* refers to any RDF resource, which the security adaptation component selects based on ontologies and policy information from the conceptual level.

When an application queries or modifies information, only some contexts and measurements are active. The access control component uses only those resources, which are active for the application in a current run-time situation. Active resources are found through the *control context* concept, which can be realized as an RDF resource. Security adaptation components define which measurement and context resources are active with RDF triplets: <*ControlContext, 'hasActive'*, *Context/Measurement*>. Determination of what resources are active is a dynamic and constantly running process, which may involve different security adaptation applications. ControlContext resources are fixed in a sense that the access control and security adaption components must know them. For instance, each smart space application, which has connected to a SIB and has an open communication socket, may have a dedicated ControlContext resource. In this case the active resources could be URIs representing end-users' identity or security level. These URIs can be resolved and activated by security adaptation component in monitor and analyze phases when the user authenticates.

5.3.2.1 Authorization Predicates

An important use case for the model is authorization over resource access. Policy predicates enabling authorization are defined in Table 10. The granularity of the model protects individual RDF resources but also semantic relationships as it is possible to control how properties of an RDF resource can be accessed. The model allows use of both allow and disallow policies. Different policies can be used in conjunction to set conditions to the authorizations (e.g. a user can access information but only if contextual requirement is met). To prevent contradicting behavior due to simultaneous use of allow and disallow policies, the proposed approach is that 'disable' policies override allow policies.

Table 10 Authorization policy predicates [1]

Predicate	Description
GetAllowedFor	Authorizes reading URI or literal value
SetAllowedFor	Authorizes modifying URI or literal value
PropertyCreationAllowedFor	Authorizes adding new URI or literal node under URI node
PropertyRemovalAllowedFor	Authorizes removal of URI or literal node from URI node
UseAsPropertyAllowedFor	Authorizes use of this node under other URI nodes
GetDisabledFor	Prevents reading URI or literal value
SetDisabledFor	Prevents modifying URI or literal value
PropertyCreationDisabledFor	Prevents adding new URI or literal node under URI node
PropertyRemovalDisabledFor	Prevents removal of URI or literal node from URI node
UseAsPropertyDisabledFor	Prevents use of this node under other URI nodes
IsAuthorizedBy	Sets node under access control and specifies authority. There may be several authorities in one broker.

The model enables efficient run-time access control. An access control component does not need to do heavy reasoning at the time applications are querying or modifying information. Instead, security adaptation analysis and planning phases can be done in advance when events, triggering adaptation, occur. An access control component needs only locate the relevant security relationships, presented with simple RDF triples, between context or measurement resources and a target sources. When a smart space application queries or modifies RDF information, the access control component checks whether there are active policies allowing or denying the action.

The inference where authorization relation is found is based on founding an *authorizing (semantic) relation* from the knowledge presented as RDF graphs. The authorization to perform particular operation can be formalized using the following notation:

$$Authorized(action, information) \Leftrightarrow$$
$$\exists Measurement/context: (rdf (Measurement/context, P_{allow}(action), information)$$
$$\wedge \exists ControlContext: rdf (ControlContext, 'hasActive', Measurement/context))$$

where P_{allow} is an allow predicate, *action* is the performed action, and *rdf* is a truth query from RDF database (whether the given RDF triple is found or not). Authorizing relation is found, if there is an active *Context* which is active and which has an authorizing relationships to requested *information*.

When the amount of active and authorizing context and measurement resources is n, the access control component must do at most 2n truth queries ('is there allow or deny relation between active resource and accessed resource') from the database to resolve the authorization of a transaction on a target. The access control component must also find active resources for each used control context resource. Implementations may further speed up this by keeping the list of control context specific active resources in cached memory.

5.3.2.2 Accounting Predicates

In addition to authorization, the model supports other real-time security control situations. Table 11 presents predicate definitions for access accounting activities, which are needed to determine authenticity or trustworthiness of information. The table defines relations for accounting predicates, which are used to log access requests, both successful and unsuccessful ones. This information is needed, e.g., when trying to detect malicious or harmful modifications and intrusions and when reasoning on which nodes may have been potentially compromised due to harmful information. The table also lists *IsSignedWith* and *HasSecurityContext* predicates, which the users can use to verify authenticity and trustworthiness of information. Trustworthiness may depend on context or measurement, which was active when information was stored .

Table 11 Predicates for access control accounting [1]

Predicate	Description
HasBeenAuthoredBy	Identifies resource's author
HasAddedPredicate	Identifies authors who have added predicates under the resource
IsSignedWith	Link to a signature proving authenticity and origin of resource
HasSecurityContext	Link to any security measurement or context resource which was active when the data was stored (needed to verify e.g. trustworthiness of data)
IsAuthorizedBy	Specifies the authority that controls security. If such relation to a known security authority is missing, access

	can be directly authorized without any other checks.
CanBeMonitored	Allows or disallows logging (e.g. due to performance or privacy)
HasBeenReadBy	Identifies contexts (users) where data has been successfully queried
HadInvalidReadAttemptBy	Identifies contexts (users) with rejected read requests
HadInvalidWriteAttemptBy	Identifies contexts (users) who have made rejected write requests

5.3.3 RIBS - A Secure Semantic Information Broker Implementation

RDF Information Bases Solution (RIBS) is a SIB, which implements the proposed RDF resource level access control solution. Communication between the RIBS and smart space agents is secured with the TLS protocol as described in Subsection 5.2.3. RIBS is able to resolve various contextual security metrics from the communications sessions. These metrics, described in Subsection 5.2.4, include information of protocol and algorithm in the current TLS session as well as information of key establishment mechanism from the certificate extension. In RIBS, TLS based end-user and certificate authentications are mapped to context resources in the RDF security model. Metered security strength information is analyzed and mapped directly to security measurement resources in the RDF security model. Further, RIBS monitors users and authors of particular information according to the presented security model.

The RIBS has been optimised to provide fast and low-power consuming information access. The implementation indexes all incoming RDF resources and thus enables RDF URIs as well as literals to be directly addressed. Relation information is stored to a to a bit cube. Bit cube has three dimensions of arrays: one dimension for subjects, one for predicates and one for objects. Figure 38 illustrates how security policies are stored to the bitcube as object-predicate plane. As security policies are presented with single bit, which is either on or off, they can be quickly checked and the amount of required memory won't increase even when the security configuration becomes more complex. Object dimension stores information on users as well as context and trust related resources. Predicate dimensions stores fixed policy relations presented in Table 10 and Table 11.

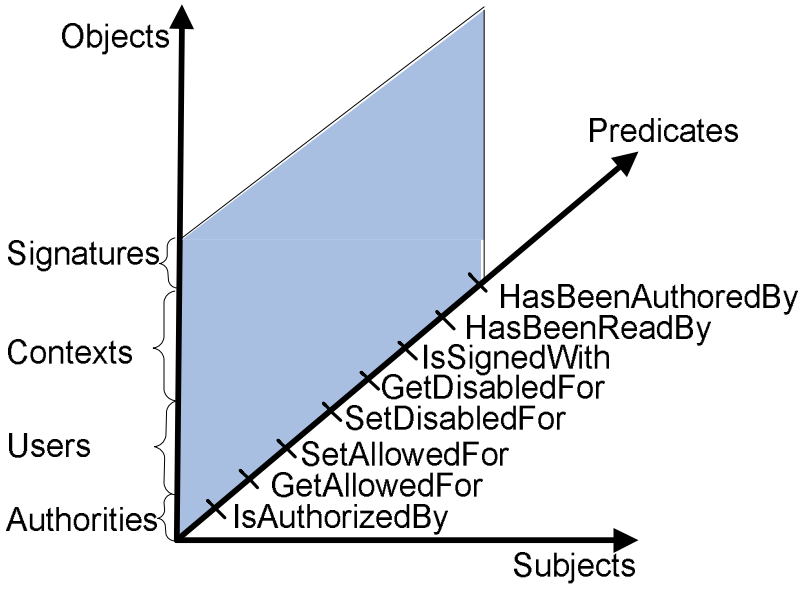

Figure 38. Security plane in RIBS subject-predicate-object bit cube [Article VI]

For each client that has joined smart space, RIBS stores the indexes of the few active rows in the object dimension. For each subject, there may be several policies i.e. object-predicate pairs active. Then when making access control checks, authorizer component of RIBS needs only to check whether the active nodes in object dimension are among active policies. For instance, when performing read action on particular subject the authorizer first checks whether this RDF resource is under access control by checking if there is '*IsAuthorizedBy'* relation between the resource and any authority object. Then we check that we have at least one '*ReadAllowedFor'* relation between the subject and any active object row for the current user. Also, we need to check that there are no '*ReadDisallowedFor'* relations.

In RIBS, all information including security policies are presented in RDF format. This enables that access control can be remotely controlled and active access control policies as well as author usage logs can be queried. The access control ontology terminology presented in the previous subsection must be known by KPs providing policies. RDF resources are accessible remotely through URI but also through internal RIBS index values. Consequently, other devices can efficiently query or add new information, including access control policies, to existing URIs, bnodes, and literal nodes. Figure 39

gives a simple example on how temperature information and access control policies are presented with RDF triples

```
<thermometerURI, hasvalue, 20>
<thermometerURI, SetAllowedFor, user:a>
<thermometerURI, GetAllowedFor, securityLevel3URI>
<thermometerURI, isAuthorizedby,securityAuthorityURI>
<temperatureURN, SetAllowedFor, securityLevel4URI>
<temperatureURN, GetAllowedFor, securityLevel3URI>
<temperatureURN, isAuthorizedby,securityAuthorityURI>
```

Figure 39. An example with information triplet and access control triplets

Due to inherent restrictions of RDF, literals do not have own unique URIs and, hence, it would be possible to add policies only to URIs (i.e. branches in RDF graphs). RIBS circumvents this limitation. Each literal has an internal address, which can be used in policies. This URN is returned for KPs when they insert RDF data to RIBS. A third party managing policies must first query URNs from RIBS or other KPs or assign policies only to branch nodes.

The RIBS has a rich set of security predicates that can be used for accounting users behaviour (See table Table 11). Bitcube enables efficient use context monitoring for smart space by attaching usage triples for RDF nodes. For instance, there are '*HasBeenAuthoredBy'* and '*HadInvalidWriteAttemptBy'* predicates, which is used to keep track of the resource accesses. To protect privacy, access to user information can be controlled so that unauthorized users cannot link monitored information to user names or certificates. Also, information specific logs are available only for those users with permission to access that particular information. Further, it is possible to define privacy policies which deny or allow logging. For instance, users may be linked with 'hasPrivacyPolicy' predicate to 'denyAuthorLogging' policy.

The trust for integrity and quality of the information is controlled with '*HasBeenAuthoredBy*' and '*IsSignedWith*' predicates. The authors and the users may have requirements for each other. E.g. the author may require that the user of the information has a certain capabilities for handling the information or belong to known trusted user group. Also, users may require that the author is either known or belongs to

a trusted group. The user may also require that authors have had sufficient security level when storing the information and hence check that predicate '*HasSecurityContext'* gives sufficient value. Consequently, consumers of information are able to resolve identities and properties of authoring nodes and make trust decisions based on that information.

5.3.3.1 Feasibility Evaluation of the Implementation

The performance of TLS based security layer was studied with two different open source TLS libraries, namely OpenSSL and GnuTLS. The performance was studied in two different platforms, particularly Windows XP (running on two processor Intel 2.40GHz laptop) and Linux/Ubuntu 9.04 (running on VMware on two processor Windows XP laptop). Latest (unoptimized and default) versions of libraries were used (OpenSSL 0.9.8 for Ubuntu and 1.0 for Windows; GnuTLS 2.6). The following mechanisms were used: TLSv1.0, RSA2048, AES-256-CBC and SHA1. For other configurations results may be different. Tests were executed in single machine with KP and RIBS processes, which were annotated to measure performance. Each test was executed ten times and average values are reported.

The performance on typical smart space operations was studied. The test case contained test triples for insert, update and query scenarios. For each unique subject, two fixed access control policy triples was generated. Typical smart space communication consists of large amount of small triplets, which may be send in larger packages. In the test set up, the insert test set consist of 425 triples, which were send in 43 SSAP packages.

Table 12 gives throughput times for the different test sets in different configurations. Additionally, some key security operations that have fixed time are listed. These include TLS handshake time, TLS library initialization time and average time of single RIBS access control check.

TABLE 12. TEST SET THROUGHPUT TIMES (MS) AND TIMES FOR SOME SECURITY OPERATIONS WITH DIFFERENT CONFIGURATIONS

	OpenSSL Windows	GnuTLS. Windows	OpenSSL Linux	GnuTLS. Linux
Insert test	46,7	119,8	91,0	128,6
Update test	32,2	106,1	153,7	152,4
Query test	48,3	239,6	93,6	142,4
Handshake	23,33	36,51	84,1	90,1
TLS init	169	157	16,2	3,80
Single AC check	0,00182	0, 00182	0, 00108	0,00108

The TLS layer causes overhead. For instance, running insert test with unsecured TCP took 19 ms in Windows and 31 ms on Linux. One reason for penalties are the heavy handshakes and library initializations. When the amount of messages increases, also the relative penalty of TLS decreases. Therefore, RIBS implementation tries to keep connections alive as long as possible. At best (with OpenSSL implementation) the penalty was between 30-35%.

Implemented access control causes also overhead and consumes memory. However, the performance penalties due to access control check are relatively small (AC check in Table 12 for a time of one check). The access control check must be done few times for each triple. Also, access control system requires additional RDF triples to represent policies. The penalty depends on the operations performed by KP:

1. SSAP Join message and authentication, which are done only once in a session, require most work. RIBS must determine, which security context objects are active. This requires e.g. looking and comparing user identifier and verification data from repository. The more users and potential context properties are available the more time is consumed.

2. Query and subscribe operations require that RIBS checks read permissions. Policies are directly linked with RDF predicates to RDF nodes. This means that RIBS checks the following truth values from the bitcube: the first check reveals whether access control is applied or not, the consecutive checks reveal whether any of the active contexts has authorization and final checks if there are deny or allow rule for active contexts. Checks must be done for each queried or subscribed triplet. However, they are fast as indexes to bitcube are resolved before hand i.e. during initialization and authentication.

3. Insert and update operations may require KPs to send additional RDF triples, which describe policies that protect these triples. For access controlled triples, the amount of additional triples is at least two. RIBS also checks that KP has write permissions.

4. Subscribed information is delivered when information is updated. For each subscribed KP, read permissions to the updated information are checked.

The average access control check time on Linux implementation was 1,08µs. In our test case, where 425 triples were inserted and 850 checks made, this means penalty of around one percent.

RIBS is optimized for environments with only few relatively static applications. Indexing all incoming RDF resources into three dimensional array consumes memory. Therefore, large scale deployments are enabled by deploying multiple brokers.

6 Towards Smart Authorization Applications

This section describes examples of access control solutions, where information from heterogenous sources can be used for authorizing access to particular resources. The section describes how secure semantic interoperability platform, presented in the previous section, can be utilized to build security applications for ubiquitous environments. Two approaches to authorization, namely role-based and popularity-based access control, will be presented to illustrate how reasoning rules can be defined for a reusable platform. Then, a piloted real-world use case, a smart door, will be described to illustrate how different components cooperate to provide smart authentication and authorization. The section is based on reasoning examples presented in Articles V and VI as well as prototyped smart door pilot implemented within the Sofia project [129].

6.1 Security Adaptation based on User Roles and Popularity of Information

The role-based access control approach (RBAC) has been recognized as a prominent model for making configuration of authorization policies usable for common users. RBAC classifies users of information according to users' roles. This access control model can be combined with models that classify the information that is accessed. Role and domain based access control issues were discussed further in Subsection 4.3.1.

The popularity-based access control protects resources according to dynamically collected information on the usage of these resources. The information on the amount of users who have either modified or accessed an RDF resource is used to classify each RDF resource into one of the nine popularity classes listed in Table 13. Information, which has several consumers or producers, is said to be popular. Information that has both consumers and producers can be said to be hot and remaining information as cold. This popularity information can be used for detecting security relevant activities. Node popularity is a dynamic measure and changes from e.g. cold to hot and from hot to cold

pose different threat situations in the smart space. The former change can be used to tighten security monitoring and access control and the latter to relieve them.

TABLE 13. RIBS NODE POPULARITY CATEGORIES [ARTICLE VI]

	No Authors	**Single Author**	**Many Authors**
No Audience	Passive	Potential Single Publisher	Potential Multi Publisher
Single Consumer	Potential Single Consumer	Point to Point Active	Multipoint toPoint Active
Wide Audience	Potential Multi Consumer	Point to Multipoint Active	Multipoint to Multipoint Active

The knowledge (on users' role, the popularity of information and the policies, authorizing roles for particular resources or controlling use of particular information) is mapped to low-level access control decisions with the reasoning rules. In the following, example rules for Answer Set Programming (ASP) [146] solver are used for deriving authorization decisions and for identifying the security threat situations. The examples illustrate that by gradually applying new rules it is possible to make a system more adaptive and self-managing to different particular situations.

A basic RBAC scenario where a rule is used to find authorized relations from the given knowledge is presented in Figure 40. Knowledge of the example contains few nodes, presenting smart space devices, which are classified to asset domains. The example contains also few roles and policies, which authorize these roles to access particular domains. The example contains also a new triplet for a new user, which is assigned to 'guards' role. Finally, the authorized rule is used to find all authorized user-device pairs. The example assumes that all authorizations have been configured by setting '*canControl*' policy relations between roles and asset domains. The example can be extended to support different ontologies and e.g. hierarchical policy models.

```
% Example RDF data:
belongs (thermostat,climate).       % Resource and its domain
belongs (lock,security).            % Resource and its domain
canControl (guards, security).      % Role-domain policy
canControl (salespersons, climate). % Role-domain policy
memberOf (new_guard,guards).        % New user and its role

% Rule for finding authorized relations:
hasAuthorization (U,N) :-
                   memberOf(U,G),belongs(N,D),canControl(G,D).
authorized (U,N) :- has Authorization(U,N).
```

Figure 40 Example RDF data and a reasoning rule for resolving authorizing relations.

In Figure 41, the example is extended by with security level based authorization (see Subsection 5.2.4). The example contains now information that the credential deployment of a new guard was based on pairing in Bluetooth 2.0. The use of Bluetooth means that the security level is 2. The example also introduces a new device, the main power switch. A new policy is defined, which requires that users must be in the level 3 to access the power switch. Finally, the reasoning rule, which finds if the user has authorization and check that the user has sufficient security level, can be presented with three lines. When an ASP solver checks the knowledge given in example, it won't find any '*authorizedTrusted*' relations. Hence, the access control enforcer, which now requires 'authorizedTrusted' relation instead of just 'hasAuthorization', does not give access permissions to anyone.

```
% New knowledge
belongs (mainPowerSwitch, security).
requiresSecurity (mainPowerSwitch, level3).
credentialsDeployedWith(new_guard, bluetoothv20).
hasSecurityGrade (bluetoothv20, level2).

% New rules:
hasTrustGrade (U, T) :- credentialsDeployedWith(U,C),
                        hasSecurityGrade(C,T).
isTrusted (U, N) :- hasTrustGrade(U,T), requiresSecurity (N,T).
authorizedTrusted(U,N) :- hasAuthorization(U,N), isTrusted(U,N).
authorized (U,N) :- authorizedTrusted(U,N)
```

Figure 41. RDF data and rule extensions for verifying trust levels required in authorization

With rules it is possible to identify different security risk situations and treat these as potential problems in smart space. This security adaptation is illustrated with a policy that uses the popularity of the RDF resource for setting the required security level. The example in Figure 42 assumes that when a multiple author situation happens, there is potentially a write conflict. The risk is higher, if the resource is i.e. if several KPs depend on its value. When this problem is detected the security level is tightened.

The figure describes example RDF data and a model where the RDF data is searched for hot nodes and for potential conflicts. A situation, where a node has many authors and it is used in two directional communication, is declared as a situation where higher security is needed. Rules *multi_author*, *conflict*, *two_directional* and *need_high_security* declare these situations respectively. When the *need_high_security* rule fires, the resource is set under access control and write access is only for the authors with sufficient security level.

```
% Example knowledge
isAuthorOf (Agent1,temperature1).
isAuthorOf (Agent2,temperature2).
isAuthorOf (Agent2,temperature1).
isUserOf (Agent3, temperature1).
isUserOf (Agent3, temperature2).
```

```
% rules for finding potential hot nodes
multi_author (V) :-       isAuthorOf(A, V),
                          isAuthorOf(B, V),
                          A != B.
two_directional (N) :-    isAuthorOf (A,N),
                          isUserOf (A,N).
conflict(V) :-            multi_author (V).
need_high_security(V) :-  two_directional (V),
                          conflict(V).

% application specific default protection rule
% sets resource under access control and raises securitylevel to 1
2 { writeAllowedFor (V, securityLev1),
accessControlled (V,authorityA)} 2
      :-  need_high_security (V).
```

Figure 42. Example of RDF data and inference rules for detecting hot nodes and conflict situation (presented in lparse format) [Article VI]

6.2 Smart Door with Adaptive Authentication and Authorization

The smart space concept as well as adaptable authentication and authorization mechanisms, were piloted with a smart door use case. In the piloted case, which can be viewed from Youtube [170], a maintenance man behind the door can ring a smart bell to reach residents from any location. A smart lock is opened either when a resident is behind the door or when a remote resident receives notification from the bell and authorizes the visitor. The case utilizes Wi-Fi for connectivity, and Near Field Communication (NFC) tags for discovering smart space. In addition, NFC is used to share temporary credentials for mobile devices that support NFC. Video cameras and smart phones are used for providing and getting information.

The case demonstrates how different phases of adaptation can be distributed to different devices and how by adding new adaptation elements, we can both increase the security level of applications as well as to make them more self-managing and, hence, user friendlier. There are several features needed in smart and context-aware access control.

The case demonstrates two kinds of user roles i.e. maintenance personnel and residents. Maintenance personnel are allowed to control a camera but not to open the lock. Residents are authorized to open door only when they have been authenticated using strong and trusted mechanisms. Residents can however lock the door using weaker mechanisms. Residents are authorized to open door only if the authentication has been performed recently. When time passes by, a re-authentication is required. When a door is opened remotely, the resident is required to use a device which has been authenticated as a trusted device.

Figure 43 presents the physical devices in the case and the deployment of different adaptation elements. RIBS, deployed to Wi-Fi access point, is the central node, which is used to share and protect information related to the door. All other components connect to it. By controlling access to information in RIBS, we control also physical security of the home. The lock device is delivered information on whether it is currently allowed to open it and whether there are people in front of the door, using this information it can adapt its physical state accordingly. Authorization to change lock's status information depends both on user's identity and role as well as on other context information. RIBS enforces that security level of the communication sessions is sufficiently strong and adapts authorizations accordingly. Security adaptation is done also in resident's terminal side, which enforces that the end-user is authenticated with appropriate mechanisms. The terminal requires users to re-authenticate if the strength of authentication is not sufficient. The system is configured with resident's terminal, which has the authority to provide security policies and credentials to RIBS. A camera and a bell provide contextual information.

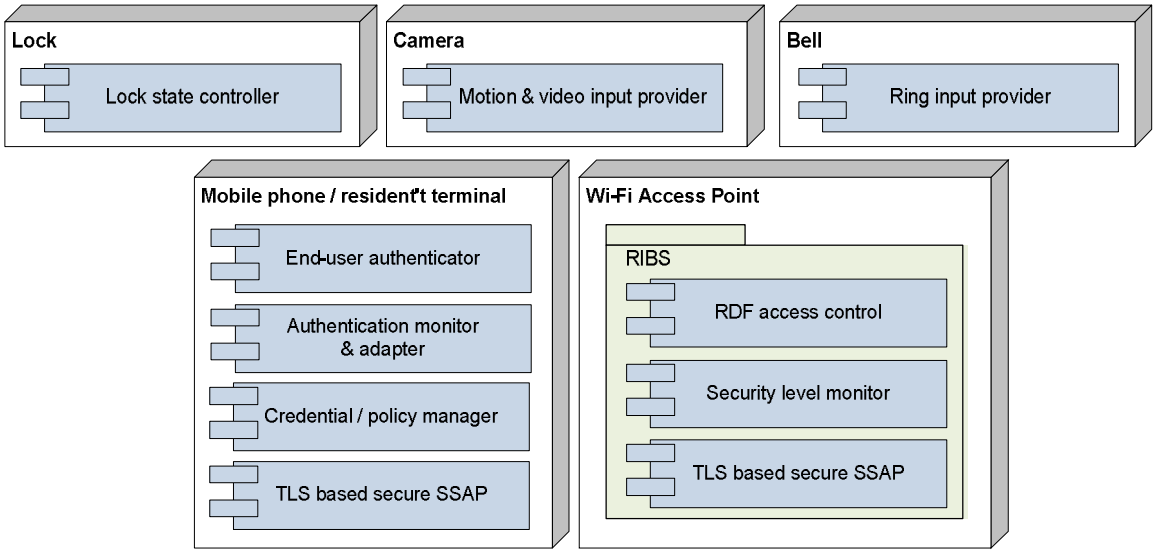

Figure 43. Deployment of security adaptation components in the smart door case

To enable that the case can be realized with devices coming from any manufacturer, we need interoperable concept definitions. Particularly, ontologies related to authentication mechanisms and their strengths are need. Also, we need ontologies to define contextual concepts related particularly for users' location and time. Necessary definitions for these concepts are defined in Information Security Measuring Ontology (ISMO) [165] and Context Ontology for Smart Spaces (CO4SS) [171]. Further, we need application specific ontologies, which specify door, lock and bell concepts as well as their relations to access control and user information, to enable specification of access control policies.

The security adaptation is performed in mobile phone and in wireless access point. Resident's mobile phone contains an implementation [172, 173] for adaptable authentication. It monitors password length and session duration metrics and analyses whether the current authentication level is sufficient for the particular information. The authentication level is determined from the length of the password and from the time passed since the last authentication. The authentication level information is then checked against the level requirements of the accessed information.

The wireless access point (RIBS) monitors metrics related to security mechanisms used in the connectivity level. These include used security protocols and their version, ciphers, authentication algorithms, and key lengths. Further, mechanisms used to establish private keys and deliver certificates as well as the terminal location (whether connections are from remote or local networks) are monitored. RIBS incorporates

security leveler and authorizer components, which analyze whether the security level of the communication session is at an acceptable level. For each session RIBS derives a security level value and for each request it analyzes whether the result value matches to the security level requirements set in the RDF security model.

The following simplified query in Figure 44 illustrate some steps of the analyses. The query is presented using SPARQL Query Language for RDF [126].

```
ASK ?user ?resource
WHERE
  { ?user hasName 'A'.
    ?user usesSecurityMechanism ?mechanism.
    ?mechanism hasSecurityContext ?securitylevel.
    ?resource readAllowedFor ?securitylevel }
```

Figure 44. Query for analyzing if used security mechanism is sufficient for reading some resource

When authorization check results an access denied situation, alternative mechanisms, which the devices are known to support and which would yield different results, are searched. The query is presented in Figure 45. If no alternatives are found, a reason for failure is recorded.

```
QUERY ?mechanisms
WHERE
  { ?user hasName 'A'.
    ?user supportsSecurityMechanism ?mechanism.
    ?resource hasIdentifier 'X'.
    ?resource readAllowedFor ?securitylevel.
    ?mechanism hasSecurityContext ?securitylevel }
```

Figure 45. Query for finding suitable authentication mechanisms to read resource ('X')

When a new user is introduced to the smart space, RIBS must be provided information enabling it to authenticate and identify this user. Further, RIBS must be configured to allow this user to access particular resources. The resources that the user is authorized to access can found by searching with queries, presented in Figure 46 and Figure 47.

```
ASK ?user ?role
WHERE
  { ?user hasName 'A'.
    ?user hasRole ?role }
```

Figure 46. Query for analyzing what roles are applicable for added user ('A')

```
QUERY ?resource
WHERE
  { ?resource readAllowedFor roleY }

QUERY ?resource
WHERE
  { ?resource writeAllowedFor roleY }
```

Figure 47. Queries for finding authorized resources for the new user with a role Y

The RIBS enforces that only authenticated and authorized users can insert and modify RDF resources. The agents in home owner's terminal are responsible of administering the authorization policies, which are stored in the RIBS for each RDF resource. Some authorization policies that the introduction of a new user could cause, according to the plan from Figure 47, are illustrated in Figure 48. There is a RDF triple for each policy as well as triples for specifying authority and thus setting the access control on for these URIs.

```
<lock_on_URI, setAllowedFor, residentA_URI>
<lock_on_URI, getAllowedFor, residentA_URI>
<lock_off_URI, setAllowedFor, residentA_URI>
<lock_off_URI, getAllowedFor, residentA_URI>
<lock_on_URI, setAllowedFor, maintenancePersonnel_URI>
<camera_URI, getAllowedFor, residentA_URI>
<camera_URI, setAllowedFor, residentA_URI>
<camera_URI, getAllowedFor, maintenancePersonnel_URI>
<camera_URI, setAllowedFor, maintenancePersonnel_URI>
<camera _URI, isAuthorizedBy, SecurityAuthority>
<lock_on_URI, isAuthorizedBy, SecurityAuthority>
<lock_off_URI, isAuthorizedBy, SecurityAuthority>
```

Figure 48. Authorizaton policies (RDF triplets), which RIBS needs to enforce security in this adaptation example

Authorization checks in RIBS follow the RDF security model, which was presented in Subsection 5.3.2. The mapping between the model and concepts of the example is illustrated in Figure 49. When RIBS authenticates a user, appropriate context and measurement resources are activated. In the use case, the maintenance man is mapped to a visitor context and the home owner is mapped to a resource, which represents owner's identity. Further, all users are mapped to security measurement resources, which describe the strength of the authentication. This mapping is done according to authentication information received from the client's terminal.

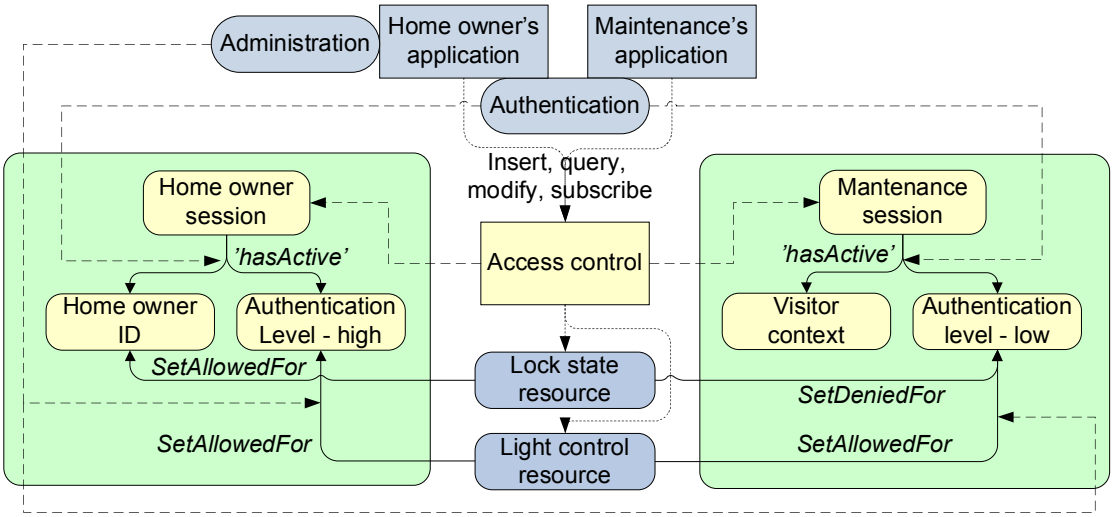

Figure 49. Examples of authorizing relations mapped to the RDF security model [1]

When users query or modify information, the RIBS checks whether these active RDF resources authorize access to requested resources. All authenticated users are given access to non-critical information inside home related e.g. to lightning. The home owner has basically access to every piece of information. However, the access to most critical information requires that the user has a sufficient authentication level (in practise this means that the user must have recently been authenticated with a trustworthy terminal).

7 Discussion

This section summarizes and discusses the work presented in the previous section. First, the section provides discussion on the significance and meaning of results. Secondly, the section presents some potential areas for future studies. The section is based on discussions and conclusions presented in Articles I-VII.

7.1 On Results

Development of smart applications, which are able to utilize information from various sources and autonomously utilize it in a manner which is the best for the current situation, has been a hot topic in the research field lately. In this research, one critical challenge is the interoperability, i.e. the ability to understand meaning of information coming from the heterogeneous network environment. This thesis searched the best ways to facilitate this development of smart applications. The research questions, presented in Subsection 1.4.1, were addressed by analyzing and exploring existing security solutions and by constructing platforms and access control solutions for secure and interoperable information sharing.

1. *How to facilitate interoperability of authentication and authorization solutions?*

In the recent decade, there has been large amount of research and standardization work to define establishment models and protocols for different connectivity technologies. This work has been aware of the diversity of personal devices. As a consequence, different physical characteristics of these devices have been utilized to provide user-friendly, secure, and cost efficient ways to introduce devices securely to each others. At the same time, when new alternative pairing models are being introduced, new interoperability issues may emerge. It is not anymore enough that a device supports one way to make the key establishment. Instead, to be a securely pairable with user's every device, a personal device should support several key establishment models. Unfortunatly, this is not always feasible. The proposed mediator for Bluetooth SSP [37] standard reflects this issue by proposing a mediator based model and protocols, which

can be used to establish keys between devices even when they would not have compatible physical interfaces.

The thesis investigated what TLS client authentication [57] would mean for the different stakeholders in the connected home ecosystem. The TLS protocol provides easily exploitable and secure mechanisms to protect communication between different devices. As it is widely available and supports flexibly various security algorithms it can be used as a common interoperability mechanism for achieving interoperability between devices with reasonable processing capabilities.

However, connectivity or network-level solutions are not enough when considering interoperability in the application level and cross-technology cases, where multiple connectivity technologies are used. In these levels, new presentation forms and architectural solutions, such as middleware protocols and platforms as well as brokers, can be used to enable cooperation over open standards. With a middleware based approaches it possible to achieve end-to-end authorization of users, software, and devices with a single solution. Use of single approach means less configuration burden and, hence, less possibilities for critical mistakes.

The smart space concept [131, 132] is a promising approach for dynamic and heterogenous ubiquitous environments. Smart spaces facilitate interoperability and ease security development in several ways. Firstly, shared knowledge of security attributes enables that it is possible to replace missing mechanisms with alternatives that are already available. Secondly, solutions are easier to update and extend as we are using open data formats and ontologies. Ontologies enable devices to share security knowledge without relying on manufacturer or standardization specific interoperability solutions. As devices using ontologies do not need to support all the defined concepts, the security adaptation solutions can more easily evolve when time passes by. Thirdly, semantic knowledge makes systems more self-configuring. Ubiquitous networks consist of large amounts of dynamic things, which emerge and leave at any time. Requiring end-users to explicitly configure their security attributes each time the environment changes is not feasible. Semantic relations and some high-level rules can be utilized to implicitly select suitable protection and select appropriate access control policies.

Fourthly, semantic technologies promise to exponentially increase the amount of knowledge that can be used in decision making. The key for realizing usable and autonomous application specific security solutions is in security and access control models, which hide the complexity of heterogeneous and rich information with abstraction.

The interoperability issues caused by the use of legacy devices and technologies can be handled with adapters. The thesis illustrated how an adapter, supporting role authentication and client authentication of TLS, can be used on low-end hardware to enable the integration of legacy devices to the connected home. A common guideline in the security field is that security should be part of product and system development from the start. However, this thesis proves that a lot can be done also to secure legacy systems. The use of TLS based approach also illustrates that implementing own middleware level security protocol is not always necessary. Middeware solutions, such as smart space access protocol (SSAP) [137] or device interconnect protocol (DIP) [140], can rely on existing and established security protocols for achieving the basic security priciples i.e. authenticity and confidentiality of communication. Higher-level solutions are then needed to enforce that the security requirements and policies set by end-users are met in controlled manner with connectivity-level approaches.

2. *How do the solutions managing heterogeneity affect to actual security level and to users' perception of security and privacy?*

The thesis gave a particular focus on measuring security level of authentication and certification authorities. These formally measured security levels can be utilized in the authorization phase to quarantee that heterogeneity of different components does not lead to compromises in the overall security level.

Security level of Authentication Mechanisms

The problem of designing ways to set up security authentication and authorization in networks of personal devices is a challenging one because it requires a balance between usability, security, and cost. The analysis in Section 2, we presented initially in [23] and then in Article I, was the first comparative analysis on the use and strength of key

establishments in standards. Since then, other researchers have surveyed the emerging device pairing methods including Uzun et al. [6], who studied usability properties of the pairing methods, and by Kumat et al. [7]. Our analysis reveals that usability improvements can be achieved without impairing the security level. New standards for Bluetooth, WUSB, and Wi-Fi, which have adopted innovative key establishment protocols, can provide effectively the same security level as the solutions based on long-passwords and symmetric cryptographic functions. Differences are mainly caused by different combinations of used physical interfaces and usability properties, however, also the protocol design was detected to affect to the achieved security properties.

The flexibility of these new proposals for smart access control introduces potential for new attacks. The novel bidding-up and bidding-down attacks against the key establishement, described in Section 2, are examples of such threats. Careful design of user dialogs may reduce the likelihood of these attacks. However, how exactly to design the user dialogs to preserve security without harming usability remains to be an open issue.

Users' Perception of Security

To assists protocol and service developers to construct and select security mechanisms, comparisons and metrics enabling systematic evaluation of security levels are needed. This thesis focused on studying the effectivity of solutions, which rely on end-user to perform additional authentication verifications. Reputation metrics provide researchers a statistical mean to quantify users' perception of trust and privacy and, hence, impact and effectiveness of security solutions. Hence, the metrics can be valuable when developing new security solutions. Also, the information on the correlation can be used by decision makers, when analyzing which security mechanisms are needed and provide enough benefits to justify the investments. As a particular example the thesis studied correlation between SSL certification, extended validation of certificates and the fine-grained metrics from Web of Trust community. The results of our-large scale HTTPS/SSL correlation analysis reinforce the doubts that extended validation in SSL certification is inefficient. The results seem to indicate that these extra security indicators, which can be easily ignored by the end-users, do not have significant impact

on authorizations decisions that end-users make. The results also revealed the differences between servers certified by different authorities. This could be interpreted as a sign that attackers tend to select particular authorizers. In the future, analyses on authorities' certification processes as well as this correlation analysis could be utilized as an incentive mechanism to introduce tighter certification practises.

The intuition was that the support for HTTPS affects to reputation in two manners: Visibility of security indicators may increase it and security warning indicators and dialogs as well as published security problems will decrease the reputation. However, service providers who are willing to invest more on HTTPS are typically also willing to invest on other factors increasing reputation. The reputation is not a result of HTTPS support. Instead, they are both results of security efforts. However, even though the correlation does not imply causality, it indicates possible causes. Future research is needed to understand, in more detail, what is the value of SSL certification and what is the value of other factors contributing to reputation.

The initial observations from the correlation study are the following:

- The results show that there is a clear correlation between HTTPS support and Web reputation. The reputation average of working SSL certificates was significantly higher than the average of servers with missing or broken certificates. Hence, it seems to pay off to have a working HTTPS support.

- The difference of reputation average between the best CA and the worst CAs was significant. Certification authorities are not typically selected from the security perspective, instead price, compatibility with browsers and easiness are likely to be more important factors. Hence, the correlation may not be used to indicate of weak certification procedures but it can be used to characterize attackers' probable selections.

- The difference between regular and extended validation certificates was insignificant. Since EV certificates are more expensive it would be likely that these service providers would had invested also in other factors contributing sites trustworthiness. For that reason we expected the trustworthiness ratings for

EV certificates to be higher. Detected correlation seems to indicate that the additional trust indicators in browsers (Figure 13 and Figure 14) are undetected by the users. This result confirms the previous small scale end-user studies that trust indicators are ignored. Hence, according to these results we could ask why to pay an extra for extended validation.

In addition to supporting development of secure solutions, the relation between SSL certification and reputation may affect to existing web security solutions. Specially, they could be usable in notary based CA selection approaches. For instance, in Convergence [75], the browser trusts only those SSL certificates which have been certified by CAs, which are accepted by particular notaries. However, it may be difficult for notaries to know which CAs to trust. Reputation gives notaries a tool, formal metric, which can be used when evaluating CAs' trustworthiness. This would act as an incentive for CAs to verify services more thoroughly, as root certificates with bad trustworthiness averages could be considered as untrusted in some browsers.

3. *How to build facilities for smart access control applications using a combination of brokers and middleware approaches?*

Networks for homes and other ubiquitous environments are growing in complexity and there is a need to increase the security but at the same time ease the effort of management they currently demand. In addition, it is becoming more important to tailor the user experience of everyday consumer electronic devices to the identity of the current user. The thesis concentrates to the view that simple-to-use authentication solutions based on open established standards and a possibility for fine-grained authorization are the main building blocks towards tackling these needs. Strong security mechanisms are difficult to make completely transparent for the end users. However, systems that can efficiently collect, share, and utilize information and available services provide a ground for building smart applications and, hence, can become more autonomous. The key enabler for efficient information sharing in dynamic environments is broker-centric store-and-subscribe architecture, which supports various intelligent agents controlling systems security behaviour. Therefore, the thesis also advocates the

use of smart space type of architectures and mechanisms to enable efficient sharing of security-relevant contextual information between devices.

Reusability is an important requirement when designing smart applications. The proposed RDF access controls model is a reusable and application-agnostic solution. The model can be used in dynamic smart space environments to provide authorization support that different applications can utilize. The model enables the information broker to enforce fine-grained access control without requiring the broker to understand and interpret end-users' high-level policies or contextual information. New smart authentication and authorization solutions can therefore be introduced to the network at any time. With the help of flexible standards for semantic information sharing and models for security reasoning, developers may more easily provide application, which will work securely in any environment. These applications are able adapt systems behaviour using contextual knowledge and rules, which are presented using application specific ontologies. Smart application examples, presented in this thesis, were constructed using standardized query interfaces and logic languages.

The thesis described design and implementation of our security solutions for smart spaces i.e. RIBS and knowledge processor side libraries. The flexible security architecture enables heterogeneous devices to share data in controlled manner and also supports policy configuration and credential deployment models, which are feasible and usable with different applications and heterogenous devices. The architecture is based on the technologies of semantic web and on proposed context-based RDF access control solution. RIBS itself also provide features for collecting contextual security information from the environment. It monitors security levels of communication sessions and tracks RDF information accesses. Consequently, it enables access control system to be adapted according to clients' security levels and popularity of information. The RIBS implementation is based on compact data structures i.e. on on bit cube. This design decision optimizes the query time that resolving a single RDF level policy requires. However, the memory consumption issues of bit cube must be addressed in the future work.

7.2 Future Research

The thesis addresses several issues, relevant in developing smart authentication and authorization applications. However, the research field is wide and in many areas the thesis only scratched the surface.

Unauthenticated key establishment models, surveyed is Section 2, enable pairing with no additional cost and with optimal usability. Hence, these models may turn out to be more preferred and more widely deployed than authenticated key agreement models. However, unauthenticated key agreement will not be sufficient for certain scenarios. One example is associating a computing device with input devices (such as keyboard or mouse), which when being malicious, can cause significant damage. Another example is a pairing of medical devices, or other similar contexts that may be subject to privacy regulation. Thus, the need for extremely inexpensive and yet secure and usable solutions for this problem remains. In-band integrity channels [46] and extracting secrets from the shared environments using existing sensors [43] seem to be promising avenues for further research.

The correlation analysis, presented in Section 3, presents an interesting and novel idea for conducting further research. The security field does not have good methods for quantifying how a security solution affects to end-users' security experience and to the trust that the end-user has towards the secured object or service. The reputation based metric was evaluated against SSL certification. In the future, other security solutions may be studied against the metric.

More studies and analysis is also needed to fully understand the causal relation between security mechanisms and end-user's perception of security. Also, in the the presented correlation analysis, between SSL certification and web reputation, needs further research to fully understand its causal meaning. Particularly, to understand all the contributing factors, it is needed to study e.g. how web service categories, application domains, and business sectors affect to servers' reputation. It may be likely that HTTPS and extended validation are typically used in more security critical services, such as banks, and that reputation evaluators valuate these services differently or more

carefully. In the future, it should be studied how the application field affects to the reputation.

Ecosystem related questions, introduced in Section 4, provide many solutions for making the security configuration and authorizations more user-friendly and secure. However, the costs and business models provide questions, which are not yet answered. Nevertheless, when issues related to interoperability can be solved, ecosystem related security services such as whitelisting of software and devices as well as outsourcing of home security, may be come more feasible and interesting. As an example, the thesis studied the idea of an access control system, which could provide a simple and effective combination. The model requires developers to categorize their offerings to standardized security domains combined with an approach where administrator users categorize the different users to roles.

There are still remaining questions and challenges related to the smart space concept, presented in Sections 5 and 6. One particular issue, related to adaptation and semantic technologies, is the computational and performance costs. Often the adaptation must be done at run-time and sometimes at real-time. The protection must be in place at once when new information emerges before it is used. Leaving resources unprotected or unavailable is not a viable option. Performance sets some limitations to the adaptation cases, which can be realized. In practice, we must make compromise when selecting what must be monitored, what information is needed in analysis and planning and how to enforce security. In this thesis, RIBS addressed performance issues with the straightforward RDF security model and an optimized broker implementation. However, the approach still does not solve the issuses related to reasoning over policy assignements. In future, more consideration must be given on how to place adaptation to those devices where the security enforcement is the most efficient. Also, more case studies are needed to understand what kind of algorithms are feasible in which situations. Furthe, the validation of the RDF security model should be done also with other broker implementations in addition to RIBS.

In the future, more large scale validation work is needed. There is a need to study what kind of adaptations and security models are feasible in smart space deployments

consisting of thousands of nodes. Smart security applications are needed particularly in applications where devices are cooperating and autonously adjusting their behaviour. For instance, the research related to the Internet of Things provides various application examples needing self-adaptation. This research may reveal new concepts included to security ontologies as well as new security models for specifying adaptation rules.

8 Conclusions

The development of authentication and authorization systems, which are able to change their behaviour according to the information from surrounding environment, requires interoperable and adaptable security mechanisms. This thesis explored solutions needed for smart interoperable authentication and authorization solutions. The thesis focused and contributed on the following distinct problem areas:

The thesis surveyed key establishment protocols and solutions for authentication and authorization. The thesis presented systematic classifications for authorization architectures and for protocols for human-mediated establishment of session keys. The relationships between different authentication protocols as well as between different authorization frameworks were shown using the presented classifications. Further, the thesis identified some challenges and new types of attacks, which are caused by the heterogenouty of standardized key establishment methods. The thesis also contributed by proposing a mediating approach for key establishment between associate incompatible devices in secure manner. Particularly, the thesis described novel protocols enabling Bluetooth SSP devices, supporting an out-of-band model, to be associated with other devices, supporting either other out-of-band or compare models.

The thesis studied the applicability of the SSL/TLS protocol based solutions in different environments i.e. in ecosystems for home and ubiquitous networks and for internet with large amount of web services and cooperating devices. To increase developers understanding on the effectiveness and impact of authentication mechanisms, the thesis proposed a novel metric idea. The proposed metric quantifies the correlation between the studied authentication mechanism and security reputation statistics. As a case study, the thesis analysed the correlation between SSL certification and web reputation.

The main focus of the thesis was to construct secure interoperability platforms for smart applications. The thesis contributed by presenting a design and implementation of an efficient security solution for semantic information broker. The broker is based on a novel RDF security model. This context-based model is an essential building block for developing fine-grained authorization solutions, which are adaptive and self-

configuring. The feasibility of the model was demonstrated with few examples of smart authorization applications. In the future, more research and larger pilot studies are needed to ensure the feasibility of complex and large-scale security adaptation applications. The secure interoperability platform is enabled by connectivity level solutions for key establishment as well as by authentication frameworks, which hide the issues caused by the heterogeneouty of devices and services.

9 References

[1] B. Lampson, M. Abadi, M. Burrows and E. Wobber. Authentication in distributed systems: theory and practice. ACM Transactions on Computer Systems 1992, November, Vol. 10, No. 4, pp. 265-310.

[2] R.S. Sandhu and P. Samarati. Access control: principle and practice. Communications Magazine, IEEE 1994, Vol. 32, No. 9, pp. 40-48.

[3] A.K. Jain, A. Ross and S. Prabhakar. An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology 2004, Vol. 14, No. 1, pp. 4-20.

[4] D. Gafurov. "A Survey of Biometric Gait Recognition: Approaches, Security and Challenges". Annual Norwegian Computer Science Conference. 2007.

[5] X. Suo, Y. Zhu and G. S. Owen. "Graphical Passwords: A Survey". Proceedings of the 21st Annual Computer Security Applications Conference. 2005. Pp. 463-472.

[6] E. Uzun, K. Karvonen and N. Asokan. "Usability Analysis of Secure Pairing Methods". Financial Cryptography and Data Security. Vol. 4886. 2007. Lecture Notes in Computer Science. Pp. 307-324.

[7] A. Kumar, N. Saxena, G. Tsudik and E. Uzun. "Caveat eptor: A comparative study of secure device pairing methods". Proceedings of the IEEE International Conference on Pervasive Computing and Communications. 2009. Pp. 1-10.

[8] M.J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad and G. D. Abowd. "Securing context-aware applications using environment roles". Proceedings of the sixth ACM symposium on Access control models and technologies. Chantilly, Virginia, United States, 2001. Pp. 10-20.

[9] H. Ko, D. Won, D. Shin, H. Choo and U. Kim. "A Semantic Context-Aware Access Control in Pervasive Environments". Computational Science and Its Applications - ICCSA 2006. Vol. 3981. 2006. Lecture Notes in Computer Science. Pp. 165-174.

[10] A. Toninelli, R. Montanari, L. Kagal and O. Lassila. "Proteus: A Semantic Context-Aware Adaptive Policy Model". Eighth IEEE International Workshop on Policies for Distributed Systems and Networks. 2007. Pp. 129-140.

[11] G. Coker, et al. Principles of remote attestation. International Journal of Information Security 2011, Vol. 10, No. 2, pp. 63-81.

[12] T. Moore and R. Clayton. "Evaluating the Wisdom of Crowds in Assessing Phishing Websites". Proceedings of the Fincancial Cryptography and Data Security. 2008. Pp. 16-30.

[13] P.H. Chia and S. J. Knapskog. "Re-Evaluating the Wisdom of Crowds in Assessing Web Security". Proceedings of the Financial Cryptography and Data Security. 2011.

[14] D.J. Cook and S.K. Das. How smart are our environments? An updated look at the state of the art. Pervasive and Mobile Computing 2007, March, Vol. 3, No. 2, pp. 53-73.

[15] J.O. Kephart and D.M. Chess. The vision of autonomic computing. Computer 2003, Vol. 36, No. 1, pp. 41-50.

[16] H. Zimmermann. OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection. IEEE Transactions on Communications 1980, April 1980, Vol. 28, No. 4, pp. 425-432.

[17] International Organization for Standardization. Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. ISO/IEC 7498-1:1994. 2nd ed. 1994.

[18] J. Farrell and G. Saloner. Standardization, Compatibility, and Innovation. The Rand journal of economics 1985, Spring, Vol. 16, No. 1, pp. 70-83.

[19] T. Berners Lee. Semantic web. Keynote speech, XML2000 conference. 2000. http://www.w3.org/2000/Talks/1206-xml2k-tbl/.

[20] World Wide Web Consortium. W3C - Security. [2010, 05/07].

[21] G.v. Bochmann and P. Mondain-Monval. Design principles for communication gateways. IEEE Journal on Selected Areas in Communications 1990, Vol. 8, No. 1, pp. 12-21.

[22] E. Kasanen, K. Lukka and A. Siitonen. The Constructive Approach in Management Accounting Research. Journal of Management Accounting Research 1993, Vol. 5, No. June, pp. 243-264.

[23] J. Suomalainen, J. Valkonen and N.Asokan. "Security Associations in Personal Networks: A Comparative Analysis". Proceedings of the 4th European Workshop on Security and Privacy in Ad-hoc and Sensor Networks. Vol. 4572. 2007. Lecture Notes in Computer Science. Pp. 43-57.

[24] C.T. Hager and S. F. Midkiff. "An analysis of Bluetooth security vulnerabilities". Wireless Communications and Networking. New Orleans, LA, USA, Vol. 3. 2003. Pp. 1825-1831.

[25] S. Wong. The evolution of wireless security in 802.11 networks: WEP, WPA and 802.11 standards. White paper. SANS Institute. 2003. http://www.leetupload.com/database/Misc/Papers/WIRELESS/SHELF/paper1109.pdf.

[26] P.R. Zimmermann. PGPfone: Pretty Good Privacy Phone Owner's Manual, Version 1.0 beta 5, Appendix C. 1996.

[27] J. Larsson. "Higher layer key exchange techniques for Bluetooth security". Open Group Conference. 2001.

[28] C. Gehrmann, C. Mitchell and K. Nyberg. Manual Authentication for Wireless Devices. RSA CryptoBytes 2004, Spring, Vol. 7, No. 1, pp. 29-37.

[29] S. Laur, N. Asokan and K. Nyberg. Efficient Mutual Data Authentication Using Manually Authenticated Strings. IACR Cryptology ePrint Archive 2005.

[30] S. Vaudenay. "Secure Communications over Insecure Channels Based on Short Authenticated Strings". Advances in Cryptology - CRYPTO 2005. Vol. 3621. 2005. Lecture Notes in Computer Science. Pp. 309-326.

[31] M. Cagalj, S. Capkun and J. Hubaux. Key Agreement in Peer-To-Peer Wireless Networks. Proceedings of the IEEE 2006, Vol. 94, No. 2, pp. 467-478.

[32] F. Stajano and R. Anderson. "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks". Proceedings of the 7th International Workshop on Security Protocols. 1999. Lecture Notes in Computer Science. Pp. 172-194.

[33] D. Balfanz, D. K. Smetters, P. Stewart and H. C. Wong. "Talking to strangers: authentication in ad-hoc wireless networks". Proceedings of the Network and Distributed System Security Symposium. 2002.

[34] J.M. McCune, A. Perrig and M. K. Reiter. "Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication". Proceedings of the 2005 IEEE Symposium on Security and Privacy. 2005. Pp. 110-124.

[35] N. Saxena, J. Ekberg, K. Kostiainen and N.Asokan. "Secure Device Pairing based on a Visual Channel (Short Paper)". Proceedings of the 2006 IEEE Symposium on Security and Privacy. 2006. Pp. 306-313.

[36] C. Soriente, G. Tsudik and E. Uzun HAPADEP: Human Assisted Pure Audio Device Pairing. Cryptology ePrint Archive, Report 2007/039, 2007.

[37] Bluetooth Special Interest Group. Bluetooth 2.1 Specifications. 2007. http://www.bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA-E5535258B056/6545/Core_V21__EDR.zip.

[38] Wi-Fi Alliance. Wi-Fi Protected Setup Specification. 2007. http://www.wi-fi.org/wifi-protected-setup/.

[39] USB Implementers Forum. Wireless Universal Serial Bus. Specification 1.1. 2010. http://www.usb.org/developers/wusb/docs/.

[40] R. Newman, S. Gavette, L. Yonge and R. Anderson. "Protecting domestic power-line communications". Proceedings of The Second Symposium on Usable Privacy and Security. 2006. Pp. 122-132.

[41] R. Newman, L. Yonge, S. Gavette and R. Anderson. "HomePlug AV Security Mechanisms". Proceedings of The International Symposium on Power Line Communications and Its Applications. 2007. Pp. 366-371.

[42] D. Dolev and A.C. Yao. On the Security of Public Key Protocols. IEEE Transactions on Information Theory 1983, Vol. 29, No. 2, pp. 198-208.

[43] A. Varshavsky, A. Scannell, A. LaMarca and E. d. Lara. "Amigo: Proximity-based Authentication of Mobile Devices". Proceedings of the Ninth International Conference on Ubiquitous Computing. Vol. 4717. 2007. Lecture Notes in Computer Science. Pp. 253-270.

[44] W. Diffie and M.E. Hellman. New Directions in Cryptography. IEEE Transactions on Information Theory 1976, Vol. IT-22, pp. 644-654.

[45] M.T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik and E. Uzun. "Loud and Clear: Human-Verifiable Authentication Based on Audio". Proceedings of the 26th IEEE International Conference on Distributed Computing Systems. 2006.

[46] M. Cagalj, J. Hubaux, S. Capkun, R. Rengaswamy, I. Tsigkogiannis and M. Srivastava. "Integrity (I) Codes: Message Integrity Protection and Authentication Over Insecure Channels". Proceedings of the 2006 IEEE Symposium on Security and Privacy. 2006. Pp. 280-294.

[47] S. Pasini and S. Vaudenay. "SAS-based Authenticated Key Agreement". Proceedings of The 9th International Workshop on Theory and Practice in Public Key Cryptography. Vol. 3958. 2006. Lecture Notes in Computer Science. Pp. 395-409.

[48] S. Laur and K. Nyberg. "Efficient Mutual Data Authentication Using Manually Authenticated Strings". The 5th International Conference on Cryptology and Network Security. Vol. 4301. 2006. Lecture Notes in Computer Science. Pp. 90-107.

[49] S.M. Bellovin and M. Merritt. "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks". Proceedings of the 1992 IEEE Symposium on Security and Privacy. 1992. Pp. 72-84.

[50] T.S. Heydt-Benjamin, D. V. Bailey, K. Fu, A. Juels and T. OHare. "Vulnerabilities in First-Generation RFID-enabled Credit Cards.". Proceedings of Eleventh International Conference on Financial Cryptography and Data Security. Vol. 4886. 2007. Lecture Notes in Computer Science. Pp. 2-14.

[51] B. Azimi-Sadjadi, A. Kiayias, A. Mercado and B. Yener. "Robust key generation from signal envelopes in wireless networks". Proceedings of the 14th ACM conference on Computer and communications security. Alexandria, Virginia, USA, 2007. Pp. 401-410.

[52] USB Implementers Forum. Wireless USB Specification. Association Models Supplement. 2006.

[53] E. Barker, W. Barker, W. Burr, W. Polk and M. Smid. NIST Special Publication 800-57. Recommendation for Key Management - Part 1: General (Revised). 2006. http://csrc.nist.gov/groups/ST/toolkit/key_management.html.

[54] T. Kivinen and M. Kojo. More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE). IETF Specification
. 2003. www.ietf.org/rfc/rfc3526.txt.

[55] J. Valkonen, A. Toivonen and K. Karvonen. "Usability Testing for Secure Device Pairing in Home Networks". UbiComp 2007 Workshop Proceedings, September 2007, Innsbruck, Austria. 2007. Pp. 457-462.

[56] A. Lakshminarayanan. "TAP - practical security protocols for wireless personal devices". Proceedings of 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. 2004.

[57] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol. Version 1.1. IETF Specification. 2006. http://www.ietf.org/rfc/4346.txt.

[58] G. Apostolopoulos, V. Peris and D. Saha. "Transport Layer Security: How much does it really cost?". In Proceedings of the IEEE INFOCOM. 1999.

[59] Cristian Coarfa, Peter Druschel, and Dan S. Wallach. Performance Analysis of TLS Web Servers. ACM Transactions on Computer Systems 2006, Vol. 24.

[60] X. Du, M. Shayman and M. Rozenblit. "Implementation and performance analysis of SNMP on a TLS/TCP base". IEEE/IFIP International Symposium on Integrated Network Management Proceedings. 2001. Pp. 453.

[61] E. Rescorla. HTTP Over TLS. IETF Specification. 2000. http://www.ietf.org/rfc/rfc2818.txt.

[62] International Telecommunication Union. ITU-T Recommendation X.509. 2008. http://www.itu.int/rec/T-REC-X.509-200811-I/en.

[63] R. Housley, W. Ford, W. Polk and D. Solo. Internet X.509 Public Key Infrastructure, Certificate and CRL Profile. IETF Standard. 1999. http://www.ietf.org/rfc/rfc2459.txt.

[64] CA/Browser Forum. Guidelines For The Issuance And Management Of Extended Validation Certificates. Version 1.3. 2010. http://www.cabforum.org/Guidelines_v1_3.pdf.

[65] G. Keizer. Hackers may have stolen over 200 SSL certificates. Computerworld. 2011. http://www.computerworld.com/s/article/9219663/Hackers_may_have_stolen_over_200_SSL_certificates.

[66] Mills, E. Comodo: Web attack broader than initially thought. CNET, 2011. Available: http://news.cnet.com/8301-27080_3-20048831-245.html.

[67] P. Eckersley and J. Burns. "An Observatory for the SSLiverse". Presentation at DEFCON 18. 2010.

[68] N. Vratonjic, J. Freudiger, V. Bindschaedler and J. Hubaux. "The Inconvenient Truth about Web Certificates". The Workshop on Economics of Information Security (WEIS). Fairfax, Virginia, USA, 2011.

[69] R. Dhamija, J. D. Tygar and M. Hearst. "Why phishing works". Proceedings of the SIGCHI conference on Human Factors in computing systems. Montreal, Quebec, Canada, 2006. CHI '06. Pp. 581-590.

[70] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri and L. F. Cranor. "Crying wolf: an empirical study of SSL warning effectiveness". Proceedings of the 18th conference on USENIX security symposium. Montreal, Canada, 2009. SSYM'09. Pp. 399-416.

[71] S. Egelman, L. F. Cranor and J. Hong. "You've been warned: an empirical study of the effectiveness of web browser phishing warnings". Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. Florence, Italy, 2008. CHI '08. Pp. 1065-1074.

[72] B.J. Fogg, et al. "What makes Web sites credible?: a report on a large quantitative study". Proceedings of the SIGCHI conference on Human factors in computing systems. Seattle, Washington, United States, 2001. CHI '01. Pp. 61-68.

[73] J. Sobey, R. Biddle, P. C. Oorschot and A. S. Patrick. "Exploring User Reactions to New Browser Cues for Extended Validation Certificates". Proceedings of the 13th European Symposium on Research in Computer Security: Computer Security. Malaga, Spain, 2008. ESORICS '08. Pp. 411-427.

[74] C. Jackson, D. R. Simon, D. S. Tan and A. Barth. "An evaluation of extended validation and picture-in-picture phishing attacks". Proceedings of the 11th

International Conference on Financial cryptography and 1st International conference on Usable Security. Scarborough, Trinidad and Tobago, 2007. FC'07/USEC'07. Pp. 281-293.

[75] M. Marlinspike. SSL and the future of authenticity. Presentation at BlackHat USA . 2011. http://www.youtube.com/watch?v=Z7Wl2FW2TcA.

[76] C. Soghoian and S. Stamm. "Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL". 3rd Hot Topics in Privacy Enhancing Technologies (HotPETs 2010). 2010. Pp. 50-61.

[77] D. Wendlandt, D. G. Andersen and A. Perrig. "Perspectives: improving SSH-style host authentication with multi-path probing". USENIX 2008 Annual Technical Conference. Boston, Massachusetts, 2008. Pp. 321-334.

[78] B. Laurie and A. Langley. Certificate Authority Transparency and Auditability. 2011. http://www.links.org/files/CertificateAuthorityTransparencyandAuditability.pdf.

[79] Google. Google Safe Browsing API. Available: https://developers.google.com/safe-browsing/. [Accessed 2012, April/6].

[80] McAfee. SiteAdvisor. Available: http://www.siteadvisor.com/. [Accessed 2012, May/6].

[81] Symantec. Norton Safe Web. Available: http://safeweb.norton.com/. [Accessed 2012, April/6].

[82] B. Cohen. "Incentives Build Robustness in BitTorrent". Proceedings of the first Workshop on the Economics of PeertoPeer systems. 2003.

[83] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy and A. Venkataramani. "Do Incentives Build Robustness in BitTorrent". 4th Symposium on Networked Systems Design and Implementation. 2007.

[84] J. Suomalainen, A. Pehrsson and J.K. Nurminen. A Secure P2P Incentive Mechanism for Mobile Devices. International Journal on Advances in Security 2009, June, Vol. 2, No. 1, pp. 42-52.

[85] D. Hausheer and B. Stiller. "PeerMint: Decentralized and Secure Accounting for Peer-to-Peer Applications". 4th International IFIP-TC6 Networking Conference. 2005. Pp. 40-52.

[86] L. Buttyan and J. Hubaux. Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks. Technical report DCS/2001/001. Swiss Federal Institute of Technology, 2001.

[87] PhishTank. Web site. Available: http://www.phishtank.com/. [Accessed 2013 January/7]

[88] Web of Trust. Web site. Available: http://www.mywot.com/. [Accessed 2012, 3/20].

[89] M. Sharifi, E. Fink and J. G. Carbonell. "Detection of Internet scam using logistic regression". Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 2011. Pp. 2168-2172.

[90] Electronic Frontier Foundation. The EFF SSL Observatory. Available: https://www.eff.org/observatory. [Accessed 2013 January/7]

[91] Alexa. Top 1,000,000 sites. Available: http://s3.amazonaws.com/alexa-static/top-1m.csv.zip. [Accessed 2011 November/1]

[92] R. Grinter, W. K. Edwards, M. Newman and N. Ducheneaut. "The work to make a home network work". Proceedings of the 9th European Conference on Computer Supported Cooperative Work. 2005.

[93] T. Rodden and S. Benford. "The evolution of buildings and implications for the design of ubiquitous domestic environments". Proceedings of the ACM Conference on Human Factors in Computing. 2003.

[94] C. Ellison. Home network security. Intel Technology Journal 2002, Vol. 6, No. 4, pp. 37-48.

[95] J.B. Brush and K. M. Inkpen. "Yours, Mine and Ours? Sharing and Use of Technology in Domestic Environments". In Proceedings of the Ubicomp. 2007.

[96] OSGi Alliance. OSGi Service Platform. Release 4. OSGi Alliance specification. 2005. http://www.osgi.org/.

[97] D. Ferraiolo and R. Kuhn. "Role-Based Access Control". The 15th National Computer Security Conference. Baltimore, Maryland, 1992. Pp. 554-563.

[98] J. Lucenius, J. Suomalainen and P. Ventola. "Implementing mobile access to heterogeneous home environment". Proceedings of the Home Oriented Informatics and Telematics (HOIT 2003). Irvine, California, USA, 2003.

[99] C. Neuman, T. Yu, S. Hartman and K. Raeburn. The Kerberos Network Authentication Service (V5). IETF Standard. 2005. http://tools.ietf.org/html/rfc4120.

[100] Microsoft. Active Directory Overview. Available: http://www.microsoft.com/en-us/server-cloud/windows-server/active-directory-overview.aspx. [Accessed 2012, 3/20].

[101] M.M. Swift, et al. "Improving the Granularity of Access Control for Windows 2000". Proceedings of the 6th ACM Symposium on Access Control Models and Technologies (SACMAT '01). 2001.

[102] Microsoft. Windows 7 Features: HomeGroup. Available: http://windows.microsoft.com/en-US/windows7/products/features/homegroup. [Accessed 2012, 3/20].

[103] Microsoft. Introducing Online Identity Integration. 2009. http://technet.microsoft.com/pt-pt/library/dd560662%28WS.10%29.aspx.

[104] Microsoft. Windows Authentication. 2011. http://technet.microsoft.com/en-us/library/cc755284%28WS.10%29.aspx.

[105] UPnP Forum. UPnP Security Ceremonies – Design Document for UPnP Device Architecture 1.0. 2003. http://www.upnp.org/.

[106] Microsoft. Devices profile for web services. Public consultation draft release. 2006. http://specs.xmlsoap.org/ws/2006/02/devprof/.

[107] S. Farrell and R. Housley. An Internet Attribute Certificate Profile for Authorization. IETF Specification. 2002. http://tools.ietf.org/html/rfc3281.

[108] P. Savolainen, E. Niemela and R. Savola. "A Taxonomy of Information Security for Service-Centric System". 33rd EUROMICRO Conference on Software Engineering and Advanced Applications. 2007. Pp. 5-12.

[109] R.M. Savola. "Towards a taxonomy for information security metrics". Proceedings of the 2007 ACM workshop on Quality of protection. Alexandria, Virginia, USA, 2007. QoP '07. Pp. 28-30.

[110] J. Zhuge and R. Yao. "Security Mechanisms for Wireless Home Network". Proceedings of the IEEE Global Telecommunications Conference (Globecom'03). 2003. Pp. 1527-1531.

[111] H. Abie, I. Dattani, M. Novkovic, J. Bigham, S. Topham and R. Savola. "GEMOM - Significant and Measurable Progress beyond the State of the Art". Third International Conference on Systems and Networks Communications. ICSNC '08.
. Sliema, Malta, 2008. Pp. 191-196.

[112] R.J. Caro Benito, D. Garrdio Marquez, P. Plaza Tron, R. Roman Castro, N. Sanz Martin and J. L. Serrano Martin. "SMEPP: A Secure Middleware For Embedded P2P". ICT-MobileSummit 2009. Santander, Spain, 2009.

[113] National institute of standards and technology (NIST). Role based access control. 2009. http://csrc.nist.gov/rbac/.

[114] M. Bacarella. Taking Advantage of Linux Capabilities. Homepage of Linux Journal, http://www.linuxjournal.com/article/5737. [Accessed 2012, April 11].

[115] C. Heath. Symbian OS Platform Security. 1st ed. Wiley, 2006.

[116] L. Badger, D. F. Sterne, D. L. Sherman, K. M. Walker and S. A. Haghighat. "Practical Domain and Type Enforcement for UNIX". Proceedings of the 1995 IEEE Symposium on Security and Privacy. 1995. SP '95. Pp. 66.

[117] K.M. Walker, D. F. Sterne, M. L. Badger, M. J. Petkac, D. L. Sherman and K. A. Oostendorp. "Confining root programs with domain and type enforcement (DTE)". Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography - Volume 6. San Jose, California, 1996. SSYM'96. Pp. 3-3.

[118] K.A. Oostendorp, et al. "Domain and type enforcement firewalls". Proceedings of the 13th Annual Computer Security Applications Conference. 1997. ACSAC '97. Pp. 122.

[119] K. Yee. "User Interaction Design for Secure Systems". Proceedings of the 4th International Conference on Information and Communications Security. Vol. 2513. 2002. Lecture Notes in Computer Science. Pp. 278-290.

[120] A. Kapadia, T. Henderson, J. J. Fielding and D. Kotz. "Virtual Walls: Protecting Digital Privacy in Pervasive Environments". The 5th International Conference on Pervasive Computing. Toronto, Ontario, Canada, Vol. LNCS 4480. 2007. Pp. 162-179.

[121] B. Schilit, N. Adams and R. Want. "Context-Aware Computing Applications". Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications. 1994. WMCSA '94. Pp. 85-90.

[122] G. Zhang and M. Parashar. "Context-aware dynamic access control for pervasive applications". Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference. 2004. Pp. 21-30.

[123] K. Kostiainen, O. Rantapuska, S. M. a. V. Roto, U. Holmstrom and K. Karvonen. "Usable Access Control inside Home Networks". Proceedings of the third IEEE International Workshop on Trust, Security, and Privacy for Ubiquitous Computing. 2007.

[124] The OpenSSL Project. OpenSSL: The Open Source Toolkit for SSL/TLS. [Accessed 2010, 11/19].

[125] S. Konno. CyberLink for C. Available: http://www.cybergarage.org/twiki/bin/view/Main/CyberLinkForC. [Accessed 2012, April/12].

[126] World Wide Web Consortium. Available: http://www.w3.org.

[127] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin and V. Terziyan. "Smart semantic middleware for the internet of things". Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics. 2008.

[128] L. Atzori, A. Iera and G. Morabito. The Internet of Things: A survey. Computer Networks 2010, Vol. 54, No. 15, pp. 2787.

[129] Sofia. Smart objects for intelligent applications. Project web page. Available: http://www.sofia-project.eu/. [Accessed 2012, April/12].

[130] SOFIA. SOFIA project, WWW site. Available: http://www.sofia-project.eu/. [Accessed 2011, November/11].

[131] J. Honkola, H. Laine, R. Brown and I. Oliver. "Cross-Domain Interoperability: A Case Study". Smart Spaces and Next Generation Wired/Wireless Networking. Vol. LNCS 5764. 2009. Pp. 22-31.

[132] Ovaska, E., Salmon Cinotti, T. & Toninelli, A. Design Principles and Practices of Interoperable Smart Spaces. Teoksessa: Anonymous Advanced Design Approaches for Emerging Software Systems: Principles, Methodologies and Tools. 2012. Ss. 18-47.

[133] M. Weiser. The computer for the 21st century. Scientific American 1991, Vol. 265, No. 3, pp. 94-104.

[134] M. Weiser. Some computer science issues in ubiquitous computing. Communications of the ACM 1993, jul, Vol. 36, No. 7, pp. 75-84.

[135] M. Satyanarayanan. Pervasive computing: vision and challenges. IEEE Personal Communications 2001, aug, Vol. 8, No. 4, pp. 10.

[136] P.T. Eugster, P.A. Felber, R. Guerraoui and A. Kermarrec. The many faces of publish/subscribe. ACM Computing Surveys 2003, jun, Vol. 35, No. 2, pp. 114-131.

[137] Smart-M3 project. Smart-M3 www-pages at SourceForge. http://sourceforge.net/projects/smart-m3/ [Accessed 2010, 05/07].

[138] J.F. Gomez-Pimpollo and R. Otaolea. Smart Objects for Intelligent Applications - ADK. IEEE Symposium on Visual Languages and Human-Centric Computing 2010, Vol. 0, pp. 267-268.

[139] A. Evesti, J. Suomalainen and E. Ovaska. Self-adaptation Approach for Smart Space Security. Manuscript.

[140] NoTAWorld. DIP - Device Interconnect Protocol http://www.notaworld.org/nota/dip. [Accessed 2013, January/7]

[141] World Wide Web Consortium. Extensible Markup Language (XML) 1.0, W3C Recommendation. http://www.w3.org/TR/xml/.

[142] World Wide Web Consortium. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation. 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[143] World Wide Web Consortium. RDF Semantics, W3C Recommendation. c. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/.

[144] Word Wide Web Consortium. RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation. http://www.w3.org/TR/rdf-schema/.

[145] World Wide Web Consortium. OWL Web Ontology Language Reference, W3C Recommendation. http://www.w3.org/TR/owl-ref/.

[146] I. Niemelä. Logic programs with stable model semantics as a constraint programming paradigm. Annals of Mathematics and Artificial Intelligence 1999, Vol. 25, No. 3, pp. 241-273.

[147] C. Baral. Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press, 2003.

[148] M. Gelfond. Answer sets. Handbook of Knowledge Representation, Elsevier, pages 285-316, 2008.

[149] World Wide Web Consortium. XML Signature Syntax and Processing (Second Edition). 2008. http://www.w3.org/TR/xmldsig-core/.

[150] Word Wide Web Consortium. XML Encryption Syntax and Processing. 2002. http://www.w3.org/TR/xmlenc-core/.

[151] OASIS. XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2. 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

[152] S. Dietzold and S. Auer. "Access control on RDF triple stores from a semantic wiki perspective". Scripting for the Semantic Web Workshop at 3rd European Semantic Web Conference. 2006.

[153] A. D'Elia, J. Honkola, D. Manzaroli and T. Cinotti. "Access Control at Triple Level: Specification and Enforcement of a Simple RDF Model to Support Concurrent Applications in Smart Environments". Smart Spaces and Next Generation Wired/Wireless Networking. Vol. 6869. 2011. Lecture Notes in Computer Science. Pp. 63-74.

[154] P. Reddivari, T. Finin and A. Joshi. "Policy based Access Control for a RDF Store". Proceedings of the Policy Management for the Web Workshop. 2005. Pp. 78-83.

[155] A. Jain and C. Farkas. "Secure resource description framework: an access control model". Proceedings of the eleventh ACM symposium on Access control models and technologies. Lake Tahoe, California, USA, 2006. SACMAT '06. Pp. 121-129.

[156] G. Flouris, I. Fundulaki, M. Michou and G. Antoniou. "Controlling Access to RDF Graphs". Future Internet - FIS 2010. Vol. 6369. 2010. Lecture Notes in Computer Science. Pp. 107-117.

[157] J. Kim, K. Jung and S. Park. "An Introduction to Authorization Conflict Problem in RDF Access Control". Knowledge-Based Intelligent Information and Engineering Systems. Vol. 5178. 2008. Lecture Notes in Computer Science. Pp. 583-592.

[158] E. Cho, Y. Kim, M. Hong and W. Cho. "Fine-Grained View-Based Access Control for RDF Cloaking". Ninth IEEE International Conference on Computer and Information Technology. 2009. Pp. 336-341.

[159] J. Bock, P. Haase, Q. Ji and R. Volz. "Benchmarking OWL Reasoners". Proceedings of the ARea2008 Workshop. 2008.

[160] K. Dentler, R. Cornet, A.t. Teije and N.d. Keizer. Comparison of reasoners for large ontologies in the OWL 2 EL profile. Semantic Web 2011, Vol. 2, No. 2, pp. 71-87.

[161] A. Khaled, M. F. Husain, L. Khan, K. W. Hamlen and B. Thuraisingham. "A Token-Based Access Control System for RDF Data in the Clouds". 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom). 2010. Pp. 104-111.

[162] J. Zhou. "Knowledge Dichotomy and Semantic Knowledge Management". Industrial Applications of Semantic Web. Vol. 188. 2005. IFIP International Federation for Information Processing. Pp. 305-316.

[163] H. Chen, F. Perich, T. Finin and A. Joshi. "SOUPA: standard ontology for ubiquitous and pervasive applications". Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference. Boston, Massachusetts, USA, 2004. Pp. 258-267.

[164] A. Herzog, N. Shahmehri and C.N. Duma H. An Ontology of Information Security. International Journal of Information Security and Privacy 2007, Vol. 1, No. 4, pp. 1-23.

[165] A. Evesti, E. Ovaska and R. Savola. "From security modelling to run-time security monitoring". European Workshop on Security in Model Driven Architecture. 2009. Pp. 33-41.

[166] V. Luukkala and I. Niemelä. "Enhancing a Smart Space with Answer Set Programming". Proceedings of the 4th International Semantic Web Rule Symposium (RuleML2010). Vol. 6403. 2010. Lecture Notes in Computer Science (LNCS). Pp. 89-103.

[167] Smart-M3 project. Sourceforge www site. Available: http://sourceforge.net/projects/smart-m3/. [Accessed 2011, August/28].

[168] The GNU Project. The GNU Transport Layer Security Library. WWW pages. http://www.gnu.org/software/gnutls/. [Accessed 2013, January/7].

[169] B.W. Lampson. Protection. ACM SIGOPS Operating Systems Review 1974, jan, Vol. 8, No. 1, pp. 18-24.

[170] VTT. Sofia VTT Smart Door. Video. Youtube: 2011. http://www.youtube.com/watch?v=a5OzMGu-BWY.

[171] S. Pantsar-Syväniemi, J. Kuusijärvi and E. Ovaska. "Supporting situation-awareness in smart spaces". First International Workshop on Self-managing Solutions for Smart Environments. 2011.

[172] A. Evesti and E. Ovaska. Ontology-Based Security Adaptation at Run-Time. IEEE International Conference on Self-Adaptive and Self-Organizing Systems 2010, pp. 204-212.

[173] A. Evesti, M. Eteläperä, J. Kiljander, J. Kuusijärvi, A. Purhonen and S. Stenudd. "Semantic Information Interoperability in Smart Spaces". Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia. 2009.